

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC970-U.S. PRO
09/752464
01/03/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出 願 年 月 日
Date of Application:

2000年 2月17日

出 願 番 号
Application Number:

特願2000-040009

出 願 人
Applicant (s):

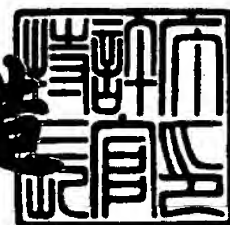
富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 9月29日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3079669

【書類名】 特許願

【整理番号】 9952105

【提出日】 平成12年 2月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00
H04L 12/56

【発明の名称】 スループットを制御する通信システムおよび方法

【請求項の数】 14

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 山中 英樹

【特許出願人】
【識別番号】 000005223
【氏名又は名称】 富士通株式会社

【代理人】
【識別番号】 100074099
【住所又は居所】 東京都千代田区二番町 8 番地 2 0 二番町ビル 3 F
【弁理士】
【氏名又は名称】 大菅 義之
【電話番号】 03-3238-0031

【選任した代理人】
【識別番号】 100067987
【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7 - 2 5 - 2 8 - 5 0 3
【弁理士】
【氏名又は名称】 久木元 彰
【電話番号】 045-573-3683

【手数料の表示】
【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 スループットを制御する通信システムおよび方法

【特許請求の範囲】

【請求項 1】 サーバとクライアントの間の通信を中継する通信システムであって、

前記サーバから前記クライアントに送信されるデータをバッファして該サーバからのデータの出力を加速することで、該サーバが該クライアントとのコネクションに対して割り当てるスループットを増大させるバッファ手段と、

前記バッファ手段に格納されたデータを前記クライアントに転送する転送手段と

を備えることを特徴とする通信システム。

【請求項 2】 サーバとクライアントの間の通信を中継する通信システムであって、

前記サーバから前記クライアントに送信されるデータを受信する受信手段と、
受信したデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換する変換手段と、

前記変換手段により変換されたデータをネットワークに送信する送信手段と
を備えることを特徴とする通信システム。

【請求項 3】 前記変換手段により変換された複数のコネクションのデータを多重化する多重化手段をさらに備え、前記送信手段は、多重化されたデータを送信することを特徴とする請求項 2 記載の通信システム。

【請求項 4】 前記クライアントに割り当てられたリソースに基づいて、アイドルリング動作を行うアイドルリング手段をさらに備え、前記送信手段は、該アイドルリング動作が終了した後にデータを送信することを特徴とする請求項 2 記載の通信システム。

【請求項 5】 前記サーバのサービス提供者に対する課金処理を行う課金手段をさらに備え、前記受信手段は、前記クライアントからの要求を前記ネットワークから受信し、該課金手段は、該クライアントからの要求が該サーバに対する要求であるか否かをチェックし、該クライアントからの要求が該サーバに対する

要求であるとき、前記送信手段は、該クライアントからの要求を該サーバに転送し、該課金手段は、該サービス提供者に対して課金することを特徴とする請求項 2 記載の通信システム。

【請求項 6】 サーバとクライアントの間の通信を中継する通信システムであって、

前記サーバから前記クライアントに送信されるデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換して得られたデータを、ネットワークから受信する受信手段と、

受信したデータのプロトコルを元のプロトコルに変換する変換手段と、

前記変換手段により変換されたデータを前記クライアントに送信する送信手段と

を備えることを特徴とする通信システム。

【請求項 7】 多重化されたデータをデマルチプレクスするデマルチプレクス手段をさらに備え、前記受信手段は、プロトコルを変換した、複数のコネクションのデータを多重化することで得られたデータを受信し、該デマルチプレクス手段は、受信したデータをデマルチプレクスし、前記変換手段は、デマルチプレクスされたデータのプロトコルを変換することを特徴とする請求項 6 記載の通信システム。

【請求項 8】 前記クライアントのユーザに対する課金処理を行う課金手段をさらに備え、前記受信手段は、前記サーバに対する要求を前記ネットワークから受信し、該課金手段は、該サーバに対する要求が該クライアントからの要求であるか否かをチェックし、該サーバに対する要求が該クライアントからの要求であるとき、前記送信手段は、該サーバに対する要求を該サーバに転送し、該課金手段は、該ユーザに対して課金することを特徴とする請求項 6 記載の通信システム。

【請求項 9】 サーバとクライアントの間の通信を制御するコンピュータのためのプログラムを記録した記録媒体であって、

前記プログラムは、

前記サーバから前記クライアントに送信されるデータをバッファして該サーバ

からのデータの出力を加速することで、該サーバが該クライアントとのコネクションに対して割り当てるスループットを増大させ、

バッファされたデータを前記クライアントに転送する
処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 0】 サーバとクライアントの間の通信を制御するコンピュータのためのプログラムを記録した記録媒体であって、

前記プログラムは、

前記サーバから前記クライアントに送信されるデータを受信し、

受信したデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換し、

変換されたデータをネットワークに送信する

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 1】 サーバとクライアントの間の通信を制御するコンピュータのためのプログラムを記録した記録媒体であって、

前記プログラムは、

前記サーバから前記クライアントに送信されるデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換して得られたデータを、ネットワークから受信し、

受信したデータのプロトコルを元のプロトコルに変換し、

変換されたデータを前記クライアントに送信する

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 2】 サーバとクライアントの間に、ネットワーク遅延を隠蔽するための仮想トンネルを構築し、

前記仮想トンネルを前記サーバと前記クライアントの間の通信バイパスとして用いて、該サーバと該クライアントの間のスループットを増大させる

ことを特徴とする通信方法。

【請求項 1 3】 前記仮想トンネルを用いた通信を利用するクライアントのユーザに対して、料金を請求することを特徴とする請求項 1 2 記載の通信方法。

【請求項 1 4】 前記仮想トンネルを用いた通信を利用するサーバのサービス提供者に対して、料金を請求することを特徴とする請求項 1 2 記載の通信方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、クライアントーサーバ間の通信、特に、大陸間、衛星中継等の遅延の大きなインターネット通信において、スループットを制御する通信システムおよびその方法に関する。

【0 0 0 2】

【従来の技術および発明が解決しようとする課題】

大陸間、衛星中継等の遠距離のインターネット通信においては、一般に、近距離の通信に比べて大きな遅延が発生し、クライアントーサーバ間の通信スループットが低下する。ここで、通信スループットとは、一定時間内に転送される情報量を指している。このような遅延の大きなネットワークにおける通信ボトルネックを解消する従来の方法としては、以下の 2 つの方法が挙げられる。

(1) 物理的に大容量のネットワークを構築する。

(2) 主サーバが提供する情報と同様のコンテンツを有するミラーサーバをクライアントの近辺に分散配置し、クライアントが最初に主サーバにアクセスした時点で、主サーバからミラーサーバにアクセスをリダイレクトする。

【0 0 0 3】

(1) の方法では、多くのユーザの通信を賄うためにバンド幅を大きくすることはできるが、物理的な距離により決まる遅延（物理的遅延）を小さくすることはできない。

【0 0 0 4】

これに対して、(2) の方法では、主サーバとクライアントの間の物理的遅延に拘束されることがなく、近距離で安価な低遅延広帯域ネットワークを実現でき

る。しかし、ミラーリングできるコンテンツに制約があり、単純なオブジェクト（データ）のダウンロードには非常に有効であるが、C G I（common gateway interface）等を用いてサーバのデータベースを更新するような通信には使えない。

【 0 0 0 5 】

また、ミラーサーバを設置するためのコストが高いという問題もある。近年の波長多重化による光ファイバを使った大陸間通信のコスト低減は著しく、バンド幅の改善だけのためにコストの高いミラーサーバを多数設置することは、割に合わなくなり始めている。

【 0 0 0 6 】

本発明の課題は、より低いコストで、クライアントーサーバ間のスループットを向上させる通信システムおよびその方法を提供することである。

【 0 0 0 7 】

【課題を解決するための手段】

図 1 は、本発明の通信システムの原理図である。

本発明の第 1 の局面において、通信システムは、バッファ手段 1 と転送手段 2 を備え、サーバ 3 とクライアント 4 の間の通信を中継する。

【 0 0 0 8 】

バッファ手段 1 は、サーバ 3 からクライアント 4 に送信されるデータをバッファしてサーバ 3 からのデータの出力を加速することで、サーバ 3 がクライアント 4 との接続に対して割り当てるスループットを増大させる。転送手段 2 は、バッファ手段 1 に格納されたデータをクライアント 4 に転送する。

【 0 0 0 9 】

バッファ手段 1 は、例えば、クライアント 4 の受信バッファより大きな容量を持ち、サーバ 3 がクライアント 4 に対して送出したデータを高速に受信する。これにより、サーバ 3 からのデータの送信速度が増大する。転送手段 2 は、バッファ手段 1 が高速に受信したデータをクライアント 4 に転送する。

【 0 0 1 0 】

サーバ 3 は、通常、ネットワーク遅延の小さな接続に対して大きなス

ループットを割り当てる制御を行うので、サーバ3の送信速度が増大するにつれて、サーバ3とクライアント4のコネクションに割り当てられるスループットが増大する。これにより、サーバ3とクライアント4の間の遅延が大きい場合でも、ミラーサーバを設置することなく、高速な通信を実現することができる。

【0011】

また、本発明の第2の局面において、通信システムは、受信手段5、変換手段6、および送信手段7を備え、サーバ3とクライアント4の間の通信を中継する。

【0012】

受信手段5は、サーバ3からクライアント4に送信されるデータを受信する。変換手段6は、受信したデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換する。送信手段7は、変換手段6により変換されたデータをネットワークに送信する。

【0013】

変換手段6は、受信手段5が受信したデータのプロトコルを別のプロトコルに変換し、送信手段7は、変換後のプロトコルに基づいて、元のプロトコルより大きなデータ量を一度にネットワークに送信する。これにより、クライアント4に向けて一度に転送されるデータ量が増大し、ミラーサーバを設置しなくても、高速な通信を実現することができる。

【0014】

また、本発明の第3の局面において、通信システムは、受信手段8、変換手段9、および送信手段10を備え、サーバ3とクライアント4の間の通信を中継する。

【0015】

受信手段8は、サーバ3からクライアント4に送信されるデータのプロトコルを、より大きなデータ量を一度に転送可能なプロトコルに変換して得られたデータを、ネットワークから受信する。変換手段9は、受信したデータのプロトコルを元のプロトコルに変換する。送信手段10は、変換手段9により変換されたデータをクライアント4に送信する。

【 0 0 1 6 】

受信手段 8 は、例えば、送信手段 7 により送信されたデータをネットワークから受信し、変換手段 9 は、受信したデータのプロトコルを変換して元のプロトコルに戻す。そして、送信手段 1 0 は、元のプロトコルに基づいて、データをクライアント 4 に送信する。これにより、クライアント 4 に向けて送信手段 7 から受信手段 8 に一度に転送されるデータ量が増大し、ミラーサーバを設置しなくても、高速な通信を実現することができる。

【 0 0 1 7 】

例えば、図 1 のバッファ手段 1 と転送手段 2 は、後述する図 2 のエージェント中継器 1 3 に対応し、図 1 の受信手段 5、変換手段 6、送信手段 7、受信手段 8、変換手段 9、および送信手段 1 0 は、後述する図 7 の受信モジュール 3 1、プロトコル変換モジュール 3 2、送信モジュール 3 4、受信モジュール 4 1、プロトコル変換モジュール 3 9、および送信モジュール 3 8 に対応する。

【 0 0 1 8 】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

大陸間通信において、コストの高いミラーサーバを分散配置することをあきらめた場合、遠距離通信の遅延によるクライアント側のスループット低下という問題が後に残される。クライアント側スループット低下の原因は、次の 3 つに大別される。

- (1) 物理的に決まる絶対遅延
- (2) 通信プロトコルによる遅延
- (3) 遅延の大きなコネクションに対するサーバのスループット割り当て低減スケジューリング

(1) の遅延は、超光速通信の実現を待たなければ解消されない。(2) の遅延は、通信プロトコルや通信の枠組みを置き換えることで、(1) に比して問題にならない程度まで小さくできる。また、(3) のスケジューリングによれば、サーバは、サーバから見た遅延が大きなコネクションに対しては、結果として小さなスループットを割り当てるような制御を行う。そこで、トータルな遅延では

なく、サーバから見たクライアントへの接続の遅延を小さくすれば、スループットを向上させることができる。

【 0 0 1 9 】

サーバから見た接続の遅延は、サーバの出力ポートへの書き込みのブロック時間として現れる。サーバの近辺という非常に低遅延な場所に中継器を置き、サーバが書き込んだデータを、中継器が大きなバッファを使って高速に読み込み、バッファに貯まったデータを、遅延の大きなクライアントとの間で緩やかに転送する。これにより、サーバにとっては、そのクライアントとの接続が非常に低遅延なものに見える。

【 0 0 2 0 】

また、通常のネットワーク通信においては、1回の転送で送信可能な最大データ量を表す通信ウィンドウが設定されており、最大通信スループットは、ウィンドウサイズと遅延時間の比（ウィンドウサイズ／遅延）により決まる。ただし、このウィンドウサイズは、送信側と受信側のバッファのサイズにより決められ、ネットワークのバンド幅は、この最大通信スループットより十分大きいものとする。

【 0 0 2 1 】

このため、遅延の大きな通信であっても、ウィンドウサイズが大きければ、ウィンドウサイズが小さくて遅延の小さな通信と同等のスループットを得ることができる。したがって、転送オブジェクトが比較的小さくて、バッファに収まる程度である場合だけでなく、ストリーミングデータのように、ほぼ同じ速度で連続的に流れてくるオブジェクトの場合でも、サーバから見た実効遅延は小さくなる。

【 0 0 2 2 】

サーバが各クライアントとの接続に割り当てるスループットは、ほとんどの場合、その接続の遅延とサーバの内部遅延のうち大きい方に反比例する。サーバの内部遅延は、ディスク I/O（入出力）、ネットワーク I/O、CPU（中央処理装置）等におけるキューの待ち時間で決まり、リクエストが多くてキューが長くなれば、それだけ内部遅延が大きくなる。

【 0 0 2 3 】

大きなサーバでは、オブジェクトを多くのディスクに分散したり、ストライピング等の方法でアクセスを並列化して、ディスク I / O の待ち時間をできるだけ小さくしている。また、ネットワーク I / O についても、高速 LAN (local area network)、WAN (wide area network) 回線を用い、複数の LAN アダプタを設ける等の工夫を施して、待ち時間を低減している。さらに、CPU のキューに関しては、複数の CPU を持つ共有記憶型計算機、複数の計算機のクラスタ等を用いて、待ち時間を低減している。

【 0 0 2 4 】

近年の急激な周辺機器の価格低減により、多くのサーバはこれらのリソースを豊富に持っているので、サーバの内部遅延はかなり小さくなってきている。その結果、クライアントへのスループット割り当ては、多くの場合、ネットワークの遅延により決まってくる。

【 0 0 2 5 】

そこで、本実施形態では、サーバの近辺にエージェント中継器を設置し、エージェント中継器が大きなバッファでオブジェクトを受信することで、サーバから見たクライアントとのネットワーク遅延を実質的に小さくする。

【 0 0 2 6 】

サーバから見た遅延を実質的に小さくすることで、合計のネットワーク遅延が同じである他のクライアントに比べて、より多くのスループットが本実施形態のシステムを利用するクライアントに割り当てられる。さらに、たとえ地球の反対側からアクセスしてくるクライアントであっても、サーバの近辺の高速ネットワークで繋がったクライアントと同等のスループットを獲得することができるようになる。

【 0 0 2 7 】

図 2 は、このような通信システムの構成図である。図 2 のシステムにおいては、Web サーバ 1 1 と Web クライアント 1 2 の間にエージェント中継器 1 3 が設けられている。サーバ 1 1 とクライアント 1 2 の間のネットワーク遅延は RT_1 であり、エージェント中継器 1 3 は、サーバ 1 1 からのネットワーク遅延が

RTT_2 で、クライアント 12 からのネットワーク遅延が RTT_3 の地点に設置されている。

【0028】

ただし、このネットワーク遅延は、送信側から受信側に信号を送信して、受信側から送信側に応答を返送するまでのラウンドトリップタイムを表し、 $RTT_1 = RTT_2 + RTT_3$ であり、エージェント中継器 13 のデータ中継オーバーヘッドは、 RTT_1 に比べて無視できるほど小さいものとする。

【0029】

サーバ 11 の内部遅延を D とし、サーバ 11 からクライアント 12 へのデータ転送のウィンドウサイズを B とすると、サーバ 11 からクライアント 12 へ直接データを転送する場合の最大ダウンロード速度は、 D と RTT_1 のうち大きい方の遅延と B との比率で決まり、 $B / \max(D, RTT_1)$ となる。 $\max(a, b)$ は、 a と b のうちの最大値を表す。通常は、 $D < RTT_1$ であるので、この最大ダウンロード速度は、 B / RTT_1 である。

【0030】

これに対して、エージェント中継器 13 が、プロキシクライアントとして、サーバ 11 とクライアント 12 の間でデータの中継を行う場合、サーバ 11 とエージェント中継器 13 の間の最大ダウンロード速度は、 $B / \max(D, RTT_2)$ となる。

【0031】

ここで、 $D < RTT_2$ となるようにエージェント中継器 13 を設置すると、サーバ 11 とエージェント中継器 13 の間の最大ダウンロード速度は、 B / RTT_2 となる。このとき、サーバ 11 から見たときの最大ダウンロード速度は、直接通信する場合に比べて、 $(B / RTT_2) / (B / RTT_1) = RTT_1 / RTT_2$ 倍となる。言い換えれば、サーバ 11 から見たときのクライアント 12 との遅延が RTT_1 から RTT_2 に減少し、サーバ 11 がクライアント 12 との接続に割り当てるスループットが RTT_1 / RTT_2 倍に増加する。

【0032】

また、エージェント中継器 13 とクライアント 12 の間の最大ダウンロード速

度は、サーバ 1 1 とエージェント中継器 1 3 の間のダウンロード速度と、エージェント中継器 1 3 とクライアント 1 2 の間のダウンロード速度のうち、遅い方の速度で決まり、 $\min(B/RTT_3, B/\max(D, RTT_2))$ となる。
 $\min(a, b)$ は、 a と b のうちの最小値を表す。

【 0 0 3 3 】

ここで、 $RTT_2 < RTT_3$ となるようにエージェント中継器 1 3 を設置し、 $D < RTT_3$ であるものとすれば、 B/RTT_3 は $B/\max(D, RTT_2)$ より小さくなる。したがって、エージェント中継器 1 3 とクライアント 1 2 の間の最大ダウンロード速度は、 B/RTT_3 となる。

【 0 0 3 4 】

このとき、クライアント 1 2 から見たときの最大ダウンロード速度は、直接通信する場合に比べて、 $(B/RTT_3) / (B/RTT_1) = RTT_1 / RTT_3$ 倍となる。 $RTT_3 < RTT_1$ であるから、直接通信の場合より、エージェント中継器 1 3 を用いた場合の方が高速になることが分かる。

【 0 0 3 5 】

このように、 $D < RTT_2 < RTT_3$ のとき、サーバ 1 1 が割り当てるスループットが RTT_1 / RTT_2 倍になり、クライアント 1 2 から見たときの最大ダウンロード速度が $RTT_1 / RTT_3 = RTT_1 / (RTT_1 - RTT_2)$ 倍になる。つまり、 RTT_2 が小さいほど、サーバ 1 1 から割り当てられるスループットが増大し、 RTT_2 が大きいほど、クライアント 1 2 に対する最大ダウンロード速度が増大する。

【 0 0 3 6 】

したがって、 RTT_2 をどの程度の値にするかは、スループットの割り当てと最大ダウンロード速度のトレードオフにより決定される。特に、 RTT_2 が D に近い値になるようにエージェント中継器 1 3 を設置すれば、直接通信する場合に比べて、スループットの割り当てを RTT_1 / D 倍程度まで増大させることができる。

【 0 0 3 7 】

このように、図 2 の通信システムによれば、サーバの持つスループット特性を

利用することで、クライアント 1 2 は、同時にアクセスする他のクライアントより有利なスループットを獲得することができる。

【 0 0 3 8 】

また、サーバとクライアントの双方を改造することなく、上述した通信プロトコルによる遅延を削減するために、サーバの近辺とクライアントの近辺にそれぞれエージェント中継器を設けるシステムも考えられる。このシステムでは、これらのエージェント中継器がプロトコル変換を行い、サーバとクライアントの間の通信のほとんどの時間をこの変換されたプロトコルで中継する。

【 0 0 3 9 】

例えば、各クライアントのリモートエージェントをサーバの近辺のエージェント中継器に仮想的に設置し、そのエージェント中継器とクライアントの近辺のエージェント中継器の間で、各クライアントとサーバの間のアプリケーションプロトコルのコネクションを多重化プロトコルに変換して中継する。このようなプロトコル変換により、2つのエージェント中継器間に巨大な通信ウィンドウを設定することで、クライアントとサーバの間のスループットが向上し、ネットワーク遅延が隠蔽される。

【 0 0 4 0 】

特に、HTTP (hypertext transfer protocol) 通信のように、1つのリクエストを送ってその応答を待つタイプの通信の場合、クライアントから送られたリクエストに対して、信頼性を保ちつつ、1ラウンドトリップタイム程度で応答オブジェクトをサーバからクライアントに転送するような、理論限界速度の中継プロトコルを実現できる。

【 0 0 4 1 】

この中継プロトコルでは、エージェント中継器間のTCP (transmission control protocol) 通信に巨大なTCPウィンドウを設定し、ほとんどの大きさのオブジェクト全体をウィンドウ内に収めて、1回の転送で送信側から受信側に送ってしまう。そして、受信側は、オブジェクトを受信したことを、1つのまとまった受信通知 (acknowledge) で送信側に知らせる。

【 0 0 4 2 】

図 3 は、このような通信システムの構成図である。図 3 のシステムにおいては、サーバ 1 1 の近辺にエージェント中継器 1 3 が設けられ、クライアント 1 2 の近辺にエージェント中継器 1 4 が設けられている。サーバ 1 1 とクライアント 1 2 の間、サーバ 1 1 とエージェント中継器 1 3 の間、エージェント中継器 1 3 とエージェント中継器 1 4 の間、エージェント中継器 1 4 とクライアント 1 2 の間のネットワーク遅延は、それぞれ、 RTT_1 、 RTT_2 、 RTT_3 、 RTT_4 である。ただし、 $RTT_1 = RTT_2 + RTT_3 + RTT_4$ であり、エージェント中継器 1 3、1 4 のデータ中継オーバーヘッドは、 RTT_1 に比べて無視できるほど小さいものとする。

【 0 0 4 3 】

通常、サーバ 1 1 が 1 つのコネクションに割り当てるネットワーク I/O バッファは、8 KB（キロバイト）程度のことが多い。この理由は、WWW（world wide web）で転送されるオブジェクトの平均的な大きさが 10～20 KB 程度であり、TCP の輻輳回避アルゴリズムにより、ウィンドウが 3 KB、6 KB、12 KB、... のように徐々に開いていくので、I/O バッファを大きくしても、実際に使われるバッファは 8 KB 程度にしかならないからである。

【 0 0 4 4 】

大きなバイナリデータを転送するとき大きなバッファは有効であるが、平均すると 8 KB を越えた領域は使われないことが多いで、過大なバッファを割り当てることは無駄である。

【 0 0 4 5 】

この I/O バッファを実効的に大きくする方法は、ウィンドウが大きく開いた状態のコネクションを多重化して、他のコネクションまたは他のクライアントと共有することである。この方法によれば、ウィンドウが大きいのので実効スループットは向上し、他のコネクションのデータ転送にも用いられるので、小さなオブジェクトの転送が多い場合でも、それらが一緒にウィンドウに詰め込まれて転送される。したがって、大きなバッファを確保しても、実質的には無駄にならない。

【 0 0 4 6 】

この方法は、2つのエージェント中継器13、14でデータを中継するため、それらのオーバーヘッドを必然的に含むことになるが、このオーバーヘッドは、ハードウェアおよびソフトウェアを高性能化すれば、それだけ小さくできるものである。これに対して、物理的な絶対遅延は、ハードウェアおよびソフトウェアの高性能化ではまったく小さくすることができず、光速を越える通信手段が現れない限り解消されない。

【0047】

このような通信ウィンドウを設定するために、エージェント中継器13、14の間に、大きなI/Oバッファを備えた信頼性のある通信トランスポートプロトコルでコネクションを張る。そして、このコネクションを多重化するプロトコルを用いて、サーバ11とクライアント12の間のアプリケーションプロトコルの通信を、このコネクション上で中継（トンネリング）する。

【0048】

エージェント中継器13、14は、トランスポート層のパケットをそのまま中継するのではなく、各アプリケーションプロトコルの性質を用いて、アプリケーションプロトコルのパケットあるいはデータを、より効率の良い多重化パケットに変換しながら、中継を行う。このとき、エージェント中継器13、14に設けられたクライアント12のエージェントは、例えば、プログラムモジュールとして実装され、アプリケーションプロトコルと多重化プロトコルの間のプロトコル変換／逆変換を行うインタフェースの役割を果たす。

【0049】

エージェント中継器13のエージェントは、クライアント12に代わって、大きなバッファでサーバ11から高速にオブジェクトを受信しながら、バッファに貯まったオブジェクトを、多重化プロトコルでエージェント中継器14のエージェントに転送する。そして、エージェント中継器14のエージェントは、多重化プロトコルで受信したオブジェクトを、クライアント12と通信するためのアプリケーションプロトコルに変換し、クライアント12に転送する。

【0050】

また、逆に、エージェント中継器14のエージェントは、クライアント12の

送信するオブジェクトを大きなバッファで受信しながら、多重化プロトコルでエージェント中継器 1 3 のエージェントに転送する。そして、エージェント中継器 1 3 のエージェントは、転送されたオブジェクトを大きなバッファで受信しながら、バッファに貯まったオブジェクトをサーバ 1 1 に高速に転送する。

【 0 0 5 1 】

この多重化プロトコルで、単純にアプリケーションプロトコルのトランスポートプロトコルパケットを中継した場合、ネットワーク遅延は中継オーバーヘッドの分だけ増大するだけであるが、上述のようなアプリケーションプロトコルレベルのプロトコル変換を用いれば、ネットワーク遅延を実質的に削減（隠蔽）することができる。ここで、トランスポートプロトコルパケットは、I P (Internet protocol) パケット等に対応する。

【 0 0 5 2 】

アプリケーションプロトコルレベルのプロトコル変換は、T C P 等の信頼性のある通信につきものであるデータ転送ウィンドウのウィンドウサイズを変換することで行われる。信頼性のある通信の最大通信速度はウィンドウサイズ／遅延で決まるので、通信速度が遅延の小さな区間と同等になるように、遅延の大きな区間のウィンドウサイズを大きく設定する。ただし、ネットワークのバンド幅は十分大きく、最大通信速度に影響を与えないものとする。

【 0 0 5 3 】

また、T C P 等では、輻輳回避のために、転送開始の直後から少しずつウィンドウサイズを大きくしていくスロースタート機構を採用している。このスロースタートによれば、アプリケーションプロトコルより下位層にある T C P のようなデータ転送プロトコルのコネクション毎にウィンドウが設定され、それが徐々に開いていく。1 つのコネクションが継続して使われ、転送エラーが発生しない場合、このコネクションのウィンドウは、パケットの転送が成功する度に倍増し、小さな初期値から指定された最大サイズまで指数関数的に開いていく。

【 0 0 5 4 】

そこで、下位のプロトコルのコネクションを継続して利用すれば、転送エラーが発生しない限り、下位コネクションのウィンドウは最大サイズまで開き切るこ

とになる。また、転送エラーが発生する場合でも、下位のプロトコルがウィンドウサイズを適切に自動調整することで、ウィンドウサイズは、転送エラーがほとんど起こらないような最大のサイズに落ち着くことになる。ここでは、下位接続の継続利用のことを、接続の多重化と呼んでいる。

【 0 0 5 5 】

上位層のプロトコルの多重化接続の使用を開始した時点では、下位接続のウィンドウが十分に開いていないので、通信は、下位接続のスロースタートの影響を受ける。しかし、ウィンドウは指数関数的に開いていくので、短時間のうちに十分に開き切り、多重化接続は、その使用時間のほとんどの間、ウィンドウが十分開いた状態の下位接続を使用することになる。

【 0 0 5 6 】

このように、ウィンドウが大きく開いた状態の接続を多重化して利用すると、常に最大のウィンドウサイズで通信が中継されるため、サーバ 1 1 とクライアント 1 2 の間の通信には、実質的にスロースタートが働かなくなる。また、多重化された接続には輻輳制御が働くので、多重化によってイタズラに輻輳を悪化させることはない。

【 0 0 5 7 】

これに対して、多重化を利用しない通常の場合、下位接続が毎回新たに設定されるため、それより上位のプロトコルの接続に、下位接続のスロースタートの影響がそのまま現れる。

【 0 0 5 8 】

T C P の場合、ウィンドウが標準的な最大サイズである 8 K B まで開くためには、1 7 K B 程度以上のオブジェクトをダウンロードしなければならない。しかし、WWW等のオブジェクトの典型的なサイズは 3 K B 程度なので、下位接続の能力を十分に利用していないが実態である。

【 0 0 5 9 】

例えば、多重化接続を利用して、3 K B のオブジェクト 1 0 個を連続してダウンロードする場合、6 個程度ダウンロードした時点で、既にウィンドウ

が 8 K B まで開き、それ以降のオブジェクトは、この開いたウィンドウのままで転送される。これに対して、多重化しない場合は、オブジェクト毎にウィンドウを新たに開く動作を行うため、最後まで 8 K B に達することはない。

【 0 0 6 0 】

このように、エージェント中継器 1 3、1 4 間の通信においては、上位層のプロトコルの多重化コネクションを使用するだけで、ほとんどの間、ウィンドウが開き切った状態になるため、スロースタートが働かない。また、下位コネクションの最大ウィンドウサイズを標準的な数値の 1 0 0 倍以上に初期化するような制御を行えば、コネクションを継続して利用しているだけで、通常ではあり得ない大きさにまでウィンドウを開くことができる。

【 0 0 6 1 】

一般に、ネットワークのバンド幅がどんなに大きくても、転送速度はウィンドウサイズ／遅延に抑制されてしまう。このため、ウィンドウサイズを十分に大きな値に設定しなければ、ネットワークリソースに大きな投資を行っても、それが十分に活用されない無駄なものになってしまう恐れがある。この観点から見たとき、上位層のプロトコルの多重化は、実質的な平均ウィンドウサイズを容易に拡大する 1 つの方法を与えている。

【 0 0 6 2 】

図 3 のシステムにおいて、サーバ 1 1 からクライアント 1 2 へ直接データを転送する場合の最大ダウンロード速度は、図 2 のシステムと同様に、 $B / \max(D, RTT_1)$ となる。

【 0 0 6 3 】

次に、エージェント中継器 1 3、1 4 により中継を行う場合、サーバ 1 1 とエージェント中継器 1 3 の間の最大ダウンロード速度は、 $B / \max(D, RTT_2)$ となる。

【 0 0 6 4 】

また、エージェント中継器 1 3、1 4 の間のデータ転送のウィンドウサイズ W は、サーバ 1 1 およびクライアント 1 2 のバッファサイズとは独立に任意に設定できるので、例えば、 B の数倍～数千倍の値に設定しておく。

【0065】

エージェント中継器13、14の間の最大ダウンロード速度は、サーバ11とエージェント中継器13の間のダウンロード速度と、エージェント中継器13、14の間のダウンロード速度のうち、遅い方の速度で決まり、 $\min(W/RTT_3, B/\max(D, RTT_2))$ となる。

【0066】

また、エージェント中継器14とクライアント12の間の最大ダウンロード速度は、サーバ11とエージェント中継器13の間のダウンロード速度と、エージェント中継器13、14の間のダウンロード速度と、エージェント中継器14とクライアント12の間のダウンロード速度のうち、最も遅い速度で決まり、 $\min(B/RTT_4, W/RTT_3, B/\max(D, RTT_2))$ となる。

【0067】

ここで、 $D < RTT_2 < RTT_3$ 、 $RTT_4 < RTT_3$ となるようにエージェント中継器13、14を設置し、 $W = B \times (RTT_3 / RTT_2)$ とすると、エージェント中継器13、14の間の最大ダウンロード速度は、 B/RTT_2 となり、エージェント中継器14とクライアント12の間の最大ダウンロード速度は、 $\min(B/RTT_4, B/RTT_2)$ となる。

【0068】

さらに、 $RTT_2 \geq RTT_4$ とすると、エージェント中継器14とクライアント12の間の最大ダウンロード速度は、 B/RTT_2 となる。このとき、クライアント12から見たときの最大ダウンロード速度は、直接通信する場合に比べて、 $(B/RTT_2) / (B/RTT_1) = RTT_1 / RTT_2$ 倍となる。 $RTT_2 < RTT_1$ であるから、直接通信の場合より、エージェント中継器13、14を用いた場合の方が高速になる。

【0069】

特に、エージェント中継器13、14を、それぞれ、サーバ11、クライアント12に隣接して設置すれば、 RTT_1 と RTT_3 が同じ程度の値になり、 RTT_2 は RTT_1 に比べてはるかに小さくなる。このとき、 W は B に比べてはるかに大きくなり、クライアント12から見たときの最大ダウンロード速度は、直接

通信の場合より、はるかに高速になる。したがって、サーバ 1 1 とクライアント 1 2 の間のスループットが飛躍的に増大する。

【 0 0 7 0 】

また、図 3 のシステムにおいて、エージェント中継器 1 3、1 4 のエージェントは、単に、スループットを可能な限り高める制御を行うだけでなく、逆に、故意にスループットを下げるような制御を行うこともできる。

【 0 0 7 1 】

例えば、エージェント中継器 1 3、1 4 が適当なタイミングでアイドリング動作を行うことで、データ転送のスロットリングが実現される。このとき、エージェント中継器 1 3、1 4 は、多数のクライアントの各々のエージェントが行うアイドリングを全体的にスケジューリングし、クライアント単位でリソース割り当てを記述することで、システムが各クライアントに割り当てるスループットを意図的に制御することができる。

【 0 0 7 2 】

図 4 は、このようなスロットリング制御を示している。まず、クライアント側のエージェント中継器 1 4 のエージェントに、適当なスループット値をリソース割り当てとして与えておく。次に、エージェント中継器 1 4 のエージェントは、この値をアイドリング時間 I に変換して、サーバ側のエージェント中継器 1 3 のエージェントに通知する。そして、エージェント中継器 1 3 のエージェントは、データ転送の度に、時間 I のアイドリング動作を実行する。

【 0 0 7 3 】

ここで、エージェント中継器 1 3 がサーバ 1 1 からデータを受信し、OK の受信通知 (acknowledge) を返送してからアイドリングを行うものとする、サーバ 1 1 とエージェント中継器 1 3 の間の最大ダウンロード速度は、 D 、 RTT_2 、および I のうち最大の時間と B との比率で決まり、 $B / \max (D, RTT_2, I)$ となる。

【 0 0 7 4 】

また、エージェント中継器 1 3、1 4 の間の最大ダウンロード速度は、 $\min (W / RTT_3, B / \max (D, RTT_2, I))$ となり、エージェント中継

器14とクライアント12の間の最大ダウンロード速度は、 $\min(B/RTT_4, W/RTT_3, B/\max(D, RTT_2, I))$ となる。

【0075】

ここで、 $W=B \times (RTT_3 / RTT_2)$ 、 $I>D$ 、 $I>RTT_2$ 、 $I>RTT_4$ とすると、エージェント中継器14とクライアント12の間の最大ダウンロード速度は、自動的に B/I となる。このとき、クライアント12から見たときの最大ダウンロード速度は、直接通信する場合に比べて、 $(B/I) / (B/RTT_1) = RTT_1 / I$ 倍となる。

【0076】

したがって、 $I<RTT_1$ であれば、直接通信の場合よりスループットが向上し、 $I>RTT_1$ であれば、直接通信の場合よりスループットが低下する。このような制御によれば、クライアント単位でスループットを増減することができ、クライアントの優先順位に応じたデータ転送が実現される。

【0077】

次に、図3のシステムに基づくサービスの例について説明する。図3のシステムによれば、2つのネットワーク間の遅延が大きいとき、これらのネットワーク間にエージェント中継器を設置することで、多重化コネクションを用いた高速な仮想トンネルが構築される。この仮想トンネルをバイパスとして用いることで、ネットワーク間の遅延を実質的に隠蔽することができる。したがって、特別料金を支払うプレミアムユーザに対して高スループットを保証するようなサービスを提供することが可能になる。

【0078】

このサービスでは、クライアント側のエージェント中継器内に、各クライアントのIPアドレスをあらかじめ登録しておき、エージェント中継器のエージェントは、接続を試みるクライアントのIPアドレスを、登録されているIPアドレスと比較することで、ユーザ認証を行う。そして、クライアントのIPアドレスが登録されていれば処理を続行し、それが登録されていなければ、処理を中断して接続を遮断する。

【0079】

このとき、クライアント側のエージェント中継器は、プロキシサーバと互換の動作を行い、サーバ側のエージェント中継器を用いた方が高速な場合にのみ、そのエージェント中継器にリクエストを中継し、その他の場合は、プロキシサーバとして直接サーバに接続する。

【 0 0 8 0 】

また、HTTP通信の場合には、IPアドレスの代わりにクッキー (cookie) を用いて、ユーザ認証を行うことができる。クッキーは、Webサイトの提供者が、Webブラウザを通じて、サイトを訪れたクライアントに一時的にデータを書き込んで保存させる技術である。クッキーには、ユーザに関する情報やサイトを訪れた日時、訪問回数等が記録される。

【 0 0 8 1 】

この場合、各ユーザにあらかじめクッキーを配布しておき、ネットワークへのアクセス時に、これをHTTPリクエストのヘッダに設定してもらう。また、クライアント側のエージェント中継器内に、各ユーザのクッキーをあらかじめ登録しておく。そして、エージェント中継器のエージェントは、各クライアントのHTTPリクエストのヘッダをチェックし、登録されたクッキーがヘッダにあれば、処理を続行し、そのようなクッキーがなければ、処理を中断して接続を遮断する。

【 0 0 8 2 】

図5は、このようなプレミアムユーザサービスを行う通信システムを示している。日本の複数のプレミアムユーザのクライアント12が米国の複数のサーバ11からサービスを受ける場合、米国、日本の適当な場所に、それぞれ、エージェント中継器13、14を設置する。一般に、エージェント中継器13、14は、それぞれ複数設置される。また、多重化コネクションを用いて、エージェント中継器13とエージェント中継器14の間に仮想トンネル21を構築し、アプリケーションプロトコルを仮想的にトンネリングさせる。

【 0 0 8 3 】

日本のプレミアムユーザは、エージェント中継器14をプロキシサーバとして設定し、米国のサーバ11にアクセスする。エージェント中継器14は、クライア

ント 1 2 からのリクエストを受け取ると、そのクライアント 1 2 用のエージェントを起動して、アクセス制御および課金制御を行う。

【 0 0 8 4 】

このエージェントは、IP アドレス、クッキー等を用いて、コネクション毎にクライアント 1 2 の認証課金を行い、そのクライアント 1 2 が事前に登録されているプレミアムユーザであるか否かをチェックする。そして、認証が成功すると、米国に設置された複数のエージェント中継器 1 3 のうち最適なエージェント中継器 1 3 にリクエストを転送する。また、認証が失敗すると、単にリクエストを廃棄して、コネクションを消去する。

【 0 0 8 5 】

エージェント中継器 1 4 のルーティングテーブルには、あらかじめ米国のサーバ 1 1 とエージェント中継器 1 3 が登録されており、このルーティングテーブルを検索することで、最適なエージェント中継器 1 3 へのルーティングが行われる。また、エージェント中継器 1 3、1 4 の間で動的にルーティング情報を交換することで、ルーティングテーブルが動的に更新される。

【 0 0 8 6 】

リクエストを受け取ったエージェント中継器 1 3 は、クライアント 1 2 用のエージェントを起動して、そのリクエストをサーバ 1 1 に渡す。サーバ 1 1 は、これと逆のルートを辿って、リクエストされたオブジェクトをクライアント 1 2 に転送する。

【 0 0 8 7 】

このサービスでは、エージェント中継器 1 4 のエージェントがクライアント認証時に同時に課金を行うことで、パケット数等に応じた従量制料金をユーザに請求することができる。また、一定期間のサービス利用料として、固定料金をユーザに請求する課金方法も考えられる。

【 0 0 8 8 】

このように、プレミアムユーザサービスでは、接続要求を送信したクライアント 1 2 を特定し、そのクライアント 1 2 と仮想トンネル 2 1 の接続の可否を決定する。接続を許可されたクライアント 1 2 には、仮想トンネル 2 1 を通じて、通信

スループットの増大というプレミアサービスが提供される。

【 0 0 8 9 】

特に、米国と日本の間のネットワーク遅延は、日本国内および米国内のそれに比べて非常に大きい。このため、プレミアユーザサービスにより、この間のバンド幅を十分に大きくすることができれば、日本のクライアント 1 2 と米国のサーバ 1 1 が直接通信する場合に比べて、1 0 倍以上の高速化が期待できる。

【 0 0 9 0 】

また、仮想トンネル 2 1 によりネットワーク間の遅延を実質的に隠蔽することで、料金を支払うサービス提供者のサーバ 1 1 とそのユーザのクライアント 1 2 の間の通信に対して、高スループットを保証するサービスを提供することも可能である。

【 0 0 9 1 】

サービス提供者がミラーサーバを世界的に分散配置するほどのコストを掛ける余裕がない場合、仮想トンネル 2 1 を利用することで、ミラーサーバを設置した場合に相当するサービスを、より安価なコストで実施することができる。

【 0 0 9 2 】

この場合、エージェント中継器 1 3、1 4 は、料金を支払ったサービス提供者のサーバ 1 1 に限定して、サーバ 1 1 とクライアント 1 2 の間の通信を中継する。より具体的には、エージェント中継器 1 3、1 4 は、接続してきたクライアント 1 2 からのオブジェクトのリクエストのうち、料金を支払った業者が登録したサーバ 1 1 に対するものだけを実際に中継し、それ以外のサーバ 1 1 へのリクエストを拒絶する。

【 0 0 9 3 】

このとき、クライアント側のエージェント中継器 1 4 は、リモートサーバ 1 1 と互換の動作を行いながら、リクエストをサーバ 1 1 上のオブジェクトに転送する。これにより、クライアント 1 2 に対して、エージェント中継器 1 4 がミラーサーバと同様の動作を行っているように見せることができ、クライアント 1 2 の設定を変更する必要がまったくない。

【 0 0 9 4 】

図6は、このようなクライアントミラーリングサービスを行う通信システムを示している。このシステムでは、サービスの契約時に登録された米国の各サーバ11に対して、それぞれ、1つずつ専用のエージェント中継器13を設置し、各エージェント中継器13と日本に設置されたエージェント中継器14をアプリケーションプロトコルの仮想トンネル21で結ぶ。一般に、エージェント中継器14は複数設置される。

【0095】

エージェント中継器14は、日本の一般ユーザに高速ミラーサーバとして開放され、そのルーティングテーブルには、各サーバ11とエージェント中継器13の組が登録されている。エージェント中継器14は、クライアント12からのリクエストを一旦受け取り、リクエストされたオブジェクトをサービスしているサーバ11をルーティングテーブル内で検索する。そして、そのサーバ11に結びつけられたエージェント中継器13を転送先として選択し、リクエストを転送する。

【0096】

リクエストを受け取ったエージェント中継器13は、エージェントを起動して、アクセス制御および課金制御を行う。エージェントは、サーバ11の認証課金を行い、転送されてきたリクエストがそのエージェント中継器13に結び付けられたサーバ11に対するものであるか否かをチェックする。

【0097】

そして、認証が成功すれば、リクエストをサーバ11に転送し、認証が失敗すれば、リクエストされたオブジェクトが存在しない旨の通知をクライアント12に返送する。これにより、プログラムの誤動作等によって転送されたリクエストを拒絶することが可能になる。

【0098】

このサービスでは、エージェント中継器13のエージェントがサーバ認証時に同時に課金を行うことで、図5のサービスと同様に、従量制料金をサービス提供者に請求することができる。また、一定期間のサービス利用料として、固定料金をサービス提供者に請求する課金方法も考えられる。

【 0 0 9 9 】

このように、クライアントミラーリングサービスでは、接続先のサーバ 1 1 を特定し、そのサーバ 1 1 とクライアント 1 2 を仮想トンネル 2 1 により接続するか否かを決定する。接続を許可されたクライアント 1 2 には、仮想トンネル 2 1 を通じて、ミラーサーバによるミラーリングと同様の効果を持つサービスが提供される。このサービスにおいても、図 5 のサービスと同様に、直接通信の場合に比べて 1 0 倍以上の高速化が期待できる。

【 0 1 0 0 】

次に、図 7 から図 1 0 までを参照しながら、エージェント中継器 1 3、1 4 上で起動されるエージェントの構成と動作を詳細に説明する。

図 7 は、エージェント中継器の構成図である。図 7 のエージェント中継器は、コンピュータを用いて構成され、受信モジュール 3 1、4 1、プロトコル変換モジュール 3 2、3 9、多重化モジュール 3 3、4 0、送信モジュール 3 4、3 8、認証課金部 3 5、スイッチモジュール 3 6、ルーティングテーブル 3 7 を備える。これらのモジュールは、例えば、プログラムモジュールとして実装され、プロトコル変換モジュール 3 2、3 9 は、クライアント 1 2 に代わって処理を行うエージェントに対応する。

【 0 1 0 1 】

まず、クライアント側のエージェント中継器 1 4 において、受信モジュール 3 1 は、クライアント 1 2 からのリクエストのアプリケーションプロトコルパケットをネットワークから受信し、受信したパケットをプロトコル変換モジュール 3 2 に転送する。

【 0 1 0 2 】

プレミアユーザサービスの場合、プロトコル変換モジュール 3 2 は、パケットの送信元の認証課金処理を認証課金部 3 5 に依頼し、処理結果を受け取る。その後、パケットをエージェントプロトコルパケットに変換して、スイッチモジュール 3 6 に渡す。ここで、エージェントプロトコルは、エージェント中継器間の多重化プロトコルに対応する。

【 0 1 0 3 】

スイッチモジュール36は、ルーティングテーブル37を検索することで転送先のルートを決定し、エージェントプロトコルパケットを多重化モジュール33に渡す。

【0104】

多重化モジュール33は、複数のコネクションのエージェントプロトコルパケットを多重化し、送信モジュール34は、多重化されたパケットを、サーバ側のエージェント中継器13に向けて、ネットワーク上に送出する。

【0105】

次に、エージェント中継器13において、受信モジュール41は、ネットワークから多重化されたパケットを受信し、多重化モジュール40は、パケットをデマルチプレクスして、個々のエージェントプロトコルパケットに分離し、スイッチモジュール36に渡す。

【0106】

スイッチモジュール36は、ルーティングテーブル37を検索することで転送先のルートを決定し、エージェントプロトコルパケットをプロトコル変換モジュール39に渡す。

【0107】

プロトコル変換モジュール39は、受け取ったエージェントプロトコルパケットをアプリケーションプロトコルパケットに変換する。さらに、クライアントミラーリングサービスの場合、送信先サーバの認証課金処理を認証課金部35に依頼し、処理結果を受け取る。その後、アプリケーションプロトコルパケットを送信モジュール38に渡す。送信モジュール38は、アプリケーションプロトコルパケットを、転送先のサーバ11に向けて、ネットワーク上に送出する。

【0108】

次に、エージェント中継器13において、受信モジュール31は、サーバ11のオブジェクトのアプリケーションプロトコルパケットをネットワークから受信し、受信したパケットをプロトコル変換モジュール32に転送する。

【0109】

プロトコル変換モジュール32は、パケットをエージェントプロトコルパケッ

トに変換して、スイッチモジュール 3 6 経由で、多重化モジュール 3 3 に渡す。多重化モジュール 3 3 は、複数のコネクションのエージェントプロトコルパケットを多重化し、送信モジュール 3 4 は、多重化されたパケットを、クライアント側のエージェント中継器 1 4 に向けて、ネットワーク上に送出する。

【 0 1 1 0 】

次に、エージェント中継器 1 4 において、受信モジュール 4 1 は、ネットワークから多重化されたパケットを受信する。多重化モジュール 4 0 は、パケットをデマルチプレクスして、個々のエージェントプロトコルパケットに分離し、スイッチモジュール 3 6 経由で、プロトコル変換モジュール 3 9 に渡す。

【 0 1 1 1 】

プロトコル変換モジュール 3 9 は、受け取ったパケットをアプリケーションプロトコルパケットに変換し、送信モジュール 3 8 に渡す。送信モジュール 3 8 は、アプリケーションプロトコルパケットを、転送先のクライアント 1 2 に向けて、ネットワーク上に送出する。

【 0 1 1 2 】

こうして、サーバ 1 1 にオブジェクトを要求したクライアント 1 2 に対して、エージェント中継器 1 3、1 4 経由で、要求されたオブジェクトがダウンロードされる。

【 0 1 1 3 】

図 8 は、認証課金部 3 5 の構成図である。図 8 の認証課金部 3 5 は、認証課金モジュール 5 1、IP データベース 5 2、および認証課金データベース 5 3 を含み、パケットに設定された IP アドレスに基づいて認証課金を行う。

【 0 1 1 4 】

まず、プレミアユーザサービスにおいて、IP データベース 5 2 には、各クライアント 1 2 の IP アドレスとユーザ識別情報（ユーザ ID）からなるレコードが格納され、認証課金データベース 5 3 には、ユーザ ID、サービスの利用期限、および処理されたパケットの数からなるレコードが格納される。

【 0 1 1 5 】

プロトコル変換モジュール 3 2 は、アプリケーションプロトコルパケットの送

信元の IP アドレスを認証課金モジュール 5 1 に渡す。認証課金モジュール 5 1 は、その IP アドレスをキーとして IP データベース 5 2 を検索し、その IP アドレスを持つレコードがあれば、そのレコードのユーザ ID を取得する。

【 0 1 1 6 】

次に、得られたユーザ ID をキーとして認証課金データベース 5 3 を検索し、そのユーザ ID を持つレコードがあれば、そのレコードの利用期限をチェックする。そして、現在の日時が利用期限内であれば、そのレコードのケット数に 1 を加算し、認証が成功したことをプロトコル変換モジュール 3 2 に通知する。

【 0 1 1 7 】

送信元の IP アドレスを持つレコードが IP データベース 5 2 にない場合、あるいは、サービスの利用期限が過ぎている場合は、認証課金モジュール 5 1 は、認証が失敗したことをプロトコル変換モジュール 3 2 に通知する。

【 0 1 1 8 】

次に、クライアントミラーリングサービスにおいて、IP データベース 5 2 には、各サーバ 1 1 の IP アドレスとサービス提供者識別情報（サービス提供者 ID）からなるレコードが格納され、認証課金データベース 5 3 には、サービス提供者 ID、サービスの利用期限、および処理されたケットの数からなるレコードが格納される。

【 0 1 1 9 】

プロトコル変換モジュール 3 9 は、エージェントプロトコルケットの送信先の IP アドレスを認証課金モジュール 5 1 に渡す。認証課金モジュール 5 1 は、プレミアムユーザサービスの場合と同様の処理を行って、認証の成功または失敗をプロトコル変換モジュール 3 9 に通知する。この場合、サービス提供者 ID がユーザ ID と同様の役割を果たす。

【 0 1 2 0 】

図 9 および図 1 0 は、図 7 のエージェント中継器の処理のフローチャートである。各モジュールは、処理すべきケットをネットワークから受信したとき、または、処理すべきケットを他のモジュールから入力されたときに、それぞれ、決められた処理を行う。

【 0 1 2 1 】

まず、スイッチモジュール 3 6 は、プロトコル変換モジュール 3 2 または多重化モジュール 4 0 からエージェントプロトコルパケットが入力されたか否かをチェックする（図 9 のステップ S 1）。パケットが入力されれば、ルーティングテーブル 3 7 を検索して、対応する転送先ルートが登録されているか否かをチェックし（ステップ S 2）、転送先ルートが登録されていれば、そのルートにパケットを転送する（ステップ S 3）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 2 】

通常、プロトコル変換モジュール 3 2 から入力されたパケットは多重化モジュール 3 3 に転送され、多重化モジュール 4 0 から入力されたパケットはプロトコル変換モジュール 3 9 に転送される。転送先ルートが登録されていなければ、パケットを廃棄し、エラー通知パケットを送信元に返送する（ステップ S 4）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 3 】

スイッチモジュール 3 6 にパケットが入力されなければ、次に、受信モジュール 4 1 は、ネットワークからエージェントプロトコルパケットを受信したか否かをチェックする（ステップ S 5）。パケットを受信すれば、そのパケットを多重化モジュール 4 0 に転送し（ステップ S 6）、多重化モジュール 4 0 は、パケットをデマルチプレクスして（ステップ S 7）、スイッチモジュール 3 6 に転送する（ステップ S 8）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 4 】

受信モジュール 4 1 がパケットを受信しなければ、次に、多重化モジュール 3 3 は、スイッチモジュール 3 6 からエージェントプロトコルパケットが入力されたか否かをチェックし（ステップ S 9）、パケットが入力されれば、そのパケットを多重化して、送信モジュール 3 4 に転送する（ステップ S 1 0）。そして、送信モジュール 3 4 は、パケットをネットワーク上に送出する。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 5 】

多重化モジュール 3 3 にパケットが入力されなければ、次に、受信モジュール 3 1 は、ネットワークからアプリケーションプロトコルパケットを受信したか否かをチェックする（図 1 0 のステップ S 1 1）。パケットを受信すれば、そのパケットをプロトコル変換モジュール 3 2 に転送する（ステップ S 1 2）。

【 0 1 2 6 】

プロトコル変換モジュール 3 2 は、パケットの送信元の認証課金処理を認証課金部 3 5 に依頼し、認証課金が成功したか否かをチェックする（ステップ S 1 3）。認証課金が成功すれば、そのパケットをエージェントプロトコルパケットに変換して、スイッチモジュール 3 6 に転送する（ステップ S 1 5）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 7 】

認証課金が失敗すれば、パケットを廃棄し、エラー通知パケットをスイッチモジュール 3 6 経由で送信元に返送する（ステップ S 1 6）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 2 8 】

受信モジュール 3 1 がパケットを受信しなければ、次に、プロトコル変換モジュール 3 9 は、スイッチモジュール 3 6 からエージェントプロトコルパケットが入力されたか否かをチェックする（ステップ S 1 7）。パケットが入力されれば、そのパケットをアプリケーションプロトコルパケットに変換し（ステップ S 1 8）、送信先の認証課金処理を認証課金部 3 5 に依頼して、認証課金が成功したか否かをチェックする（ステップ S 1 9）。

【 0 1 2 9 】

認証課金が成功すれば、パケットを送信モジュール 3 8 に転送し、送信モジュール 3 8 は、パケットをネットワーク上に送出する（ステップ S 2 0）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 3 0 】

認証課金が失敗すれば、パケットを廃棄し、エラー通知パケットをスイッチモジュール 3 6 経由で送信元に返送する（ステップ S 2 1）。その後、ステップ S 1 以降の処理が繰り返される。

【 0 1 3 1 】

図 1 1 は、図 8 の認証課金モジュール 5 1 の処理のフローチャートである。まず、認証課金モジュール 5 1 は、プロトコル変換モジュール 3 2 または 3 9 から認証課金が要求されたか否かをチェックする（ステップ S 3 1）。

【 0 1 3 2 】

プロトコル変換モジュール 3 2 から認証課金が要求されれば、その要求に含まれる送信元の IP アドレスをキーとして IP データベース 5 2 を検索し、その IP アドレスを持つレコードがあるか否かをチェックする（ステップ S 3 2）。

【 0 1 3 3 】

そのようなレコードがあれば、次に、そのレコードに含まれるユーザ ID をキーとして認証課金データベース 5 3 を検索し、そのユーザ ID を持つレコードの処理を行って、処理結果をチェックする（ステップ S 3 3）。この処理では、レコードに含まれる利用期限をチェックし、現在の日時が利用期限内であれば、そのレコードのパケット数に 1 を加算する。

【 0 1 3 4 】

このような処理が成功すれば、認証が成功したことをプロトコル変換モジュール 3 2 に通知する（ステップ S 3 4）。その後、ステップ S 3 1 以降の処理が繰り返される。

【 0 1 3 5 】

ステップ S 3 2 において対応するレコードがない場合、または、ステップ S 3 3 において対応するレコードの処理が失敗した場合は、認証が失敗したことをプロトコル変換モジュール 3 2 に通知する（ステップ S 3 5）。ステップ S 3 3 においては、例えば、対応するレコードがないか、または利用期限が過ぎている場合に、レコードの処理が失敗したものと判断される。その後、ステップ S 3 1 以降の処理が繰り返される。

【 0 1 3 6 】

また、ステップ S 3 1 において、プロトコル変換モジュール 3 9 から認証課金が要求されれば、その要求に含まれる送信先の IP アドレスをキーとして IP データベース 5 2 を検索する。そして、ユーザ ID の代わりにサービス提供者 ID

を取得して、同様の処理を行い、結果をプロトコル変換モジュール 3 9 に通知する。

【0137】

ところで、図 3 のサーバ 1 1、クライアント 1 2、エージェント中継器 1 3、1 4 は、例えば、図 1 2 に示すような情報処理装置（コンピュータ）を用いて構成することができる。図 1 2 の情報処理装置は、CPU（中央処理装置）6 1、メモリ 6 2、入力装置 6 3、出力装置 6 4、外部記憶装置 6 5、媒体駆動装置 6 6、およびネットワーク接続装置 6 7 を備え、それらはバス 6 8 により互いに接続されている。

【0138】

メモリ 6 2 は、例えば、ROM（read only memory）、RAM（random access memory）等を含み、処理に用いられるプログラムとデータを格納する。図 7 のルーティングテーブル 3 7 と、図 8 の IP データベース 5 2 および認証課金データベース 5 3 は、例えば、メモリ 6 2 に格納される。CPU 6 1 は、メモリ 6 2 を利用してプログラムを実行することにより、必要な処理を行う。

【0139】

図 7 の受信モジュール 3 1、4 1、プロトコル変換モジュール 3 2、3 9、多重化モジュール 3 3、4 0、送信モジュール 3 4、3 8、およびスイッチモジュール 3 6 と、図 8 の認証課金モジュール 5 1 は、プログラムにより記述されたソフトウェアコンポーネントとしてメモリ 6 2 に格納される。

【0140】

入力装置 6 3 は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、オペレータ（ユーザ、サービス提供者、管理者等）からの指示や情報の入力に用いられる。出力装置 6 4 は、例えば、ディスプレイ、プリンタ、スピーカ等であり、オペレータへの問い合わせや処理結果の出力に用いられる。

【0141】

外部記憶装置 6 5 は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク（magneto-optical disk）装置、テープ装置等である。情報処理装置は、この外部記憶装置 6 5 に、上述のプログラムとデータを保存しておき、必要に応

じて、それらをメモリ 6 2 にロードして使用する。

【 0 1 4 2 】

媒体駆動装置 6 6 は、可搬記録媒体 6 9 を駆動し、その記録内容にアクセスする。可搬記録媒体 6 9 としては、メモリカード、フロッピーディスク、CD-ROM (compact disk read only memory)、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。オペレータは、この可搬記録媒体 6 9 に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ 6 2 にロードして使用する。

【 0 1 4 3 】

ネットワーク接続装置 6 7 は、通信ネットワークへの接続に用いられる。情報処理装置は、上述のプログラムとデータをネットワーク接続装置 6 7 を介して他の装置から受け取り、必要に応じて、それらをメモリ 6 2 にロードして使用する。

【 0 1 4 4 】

図 1 3 は、図 1 2 の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体 6 9 や外部のデータベース 7 0 に保存されたプログラムとデータは、メモリ 6 2 にロードされる。そして、CPU 6 1 は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【 0 1 4 5 】

以上説明した実施形態では、エージェント中継器 1 3、1 4 をサーバ 1 1、クライアント 1 2 とは別に設けているが、これらをそれぞれサーバ 1 1、クライアント 1 2 に組み込んでもよい。実際、サーバ側のエージェント中継器 1 3 は、サーバ 1 1 と一体となって運用されることが望ましい。

【 0 1 4 6 】

また、エージェント中継器 1 3、1 4 を複数のプログラムモジュールにより構成する代わりに、ハードウェアにより構成することもできる。図 1 4 は、このようなエージェント中継器の構成図である。図 1 4 のエージェント中継器は、受信機 7 1、8 1、プロトコル変換器 7 2、7 9、多重化装置 7 3、8 0、送信機 7

4、78、認証課金部75、スイッチ76、ルーティングテーブル77を備える。各構成要素は、ハードウェア回路として実装され、図7の各モジュールと同様の動作を行う。

【0147】

今日のWeb上では、プレゼンテーション効果、ユーザのプロファイリング、Webページの動的カスタム化等のため、コンテンツが静的なオブジェクトから動的なオブジェクトに移行している。このため、転送されるデータ量が増大し、キャッシュサーバ等によるトラフィック削減の効果が著しく低下している。

【0148】

また、トラフィックを削減するために、ミラーサーバを単に世界中に分散配置したのでは、ユーザプロファイルを一元管理することができない。このように、Webのトレンドは、実質的なトラフィックにおいて、情報の発信元であるサーバへの流れが増大する方向にある。

【0149】

本発明の通信システムによれば、キャッシュやミラーオブジェクトを用意することなく、サーバとクライアントの間の通信スループットを改善することができる。したがって、このシステムは、上述のようなWebのトレンドにマッチし、今後益々有効性を増すものと考えられる。

【0150】

また、本発明の通信システムによれば、トンネリングの方法を用いて、インターネットのIPネットワーク上に、実質的に通信速度が速いバイパスが作り出される。このバイパスは、ユーザに対してプレミアサービスを提供し、サービス提供者に対しては、ユーザを引きつけ、自社のサービスを他社と差別化する手段を提供する。

【0151】

【発明の効果】

本発明によれば、大陸間、衛星中継等の遅延の大きなインターネット通信において、多数のミラーサーバを分散配置することなく、クライアントーサーバ間のスループットを向上させることができる。

【図面の簡単な説明】

【図 1】

本発明の通信システムの原理図である。

【図 2】

第 1 の通信システムの構成図である。

【図 3】

第 2 の通信システムの構成図である。

【図 4】

スロットリング制御を示す図である。

【図 5】

プレミアムユーザサービスを示す図である。

【図 6】

クライアントミラーリングサービスを示す図である。

【図 7】

第 1 のエージェント中継器の構成図である。

【図 8】

認証課金部の構成図である。

【図 9】

エージェント中継器の処理のフローチャート（その 1）である。

【図 10】

エージェント中継器の処理のフローチャート（その 2）である。

【図 11】

認証課金処理のフローチャートである。

【図 12】

情報処理装置の構成図である。

【図 13】

記録媒体を示す図である。

【図 14】

第 2 のエージェント中継器の構成図である。

【符号の説明】

- 1 バッファ手段
- 2 転送手段
- 3 サーバ
- 4 クライアント
- 5、8 受信手段
- 6、9 変換手段
- 7、10 送信手段
- 11 Webサーバ
- 12 Webクライアント
- 13、14 エージェント中継器
- 21 仮想トンネル
- 31、41 受信モジュール
- 32、39 プロトコル変換モジュール
- 33、40 多重化モジュール
- 34、38 送信モジュール
- 35、75 認証課金部
- 36 スイッチモジュール
- 37、77 ルーティングテーブル
- 51 認証課金モジュール
- 52 IPデータベース
- 53 認証課金データベース
- 61 CPU
- 62 メモリ
- 63 入力装置
- 64 出力装置
- 65 外部記憶装置
- 66 媒体駆動装置
- 67 ネットワーク接続装置

6 8 バス

6 9 可搬記録媒体

7 0 データベース

7 1、8 1 受信機

7 2、7 9 プロトコル変換器

7 3、8 0 多重化装置

7 4、7 8 送信機

7 6 スイッチ

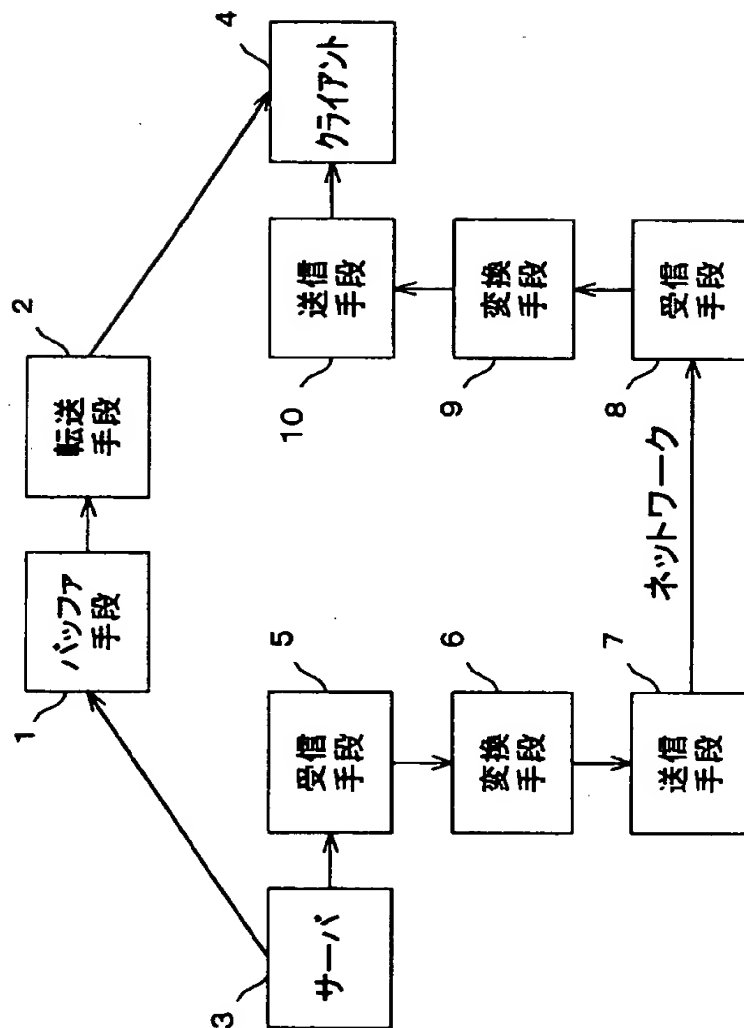
//

【書類名】

図面

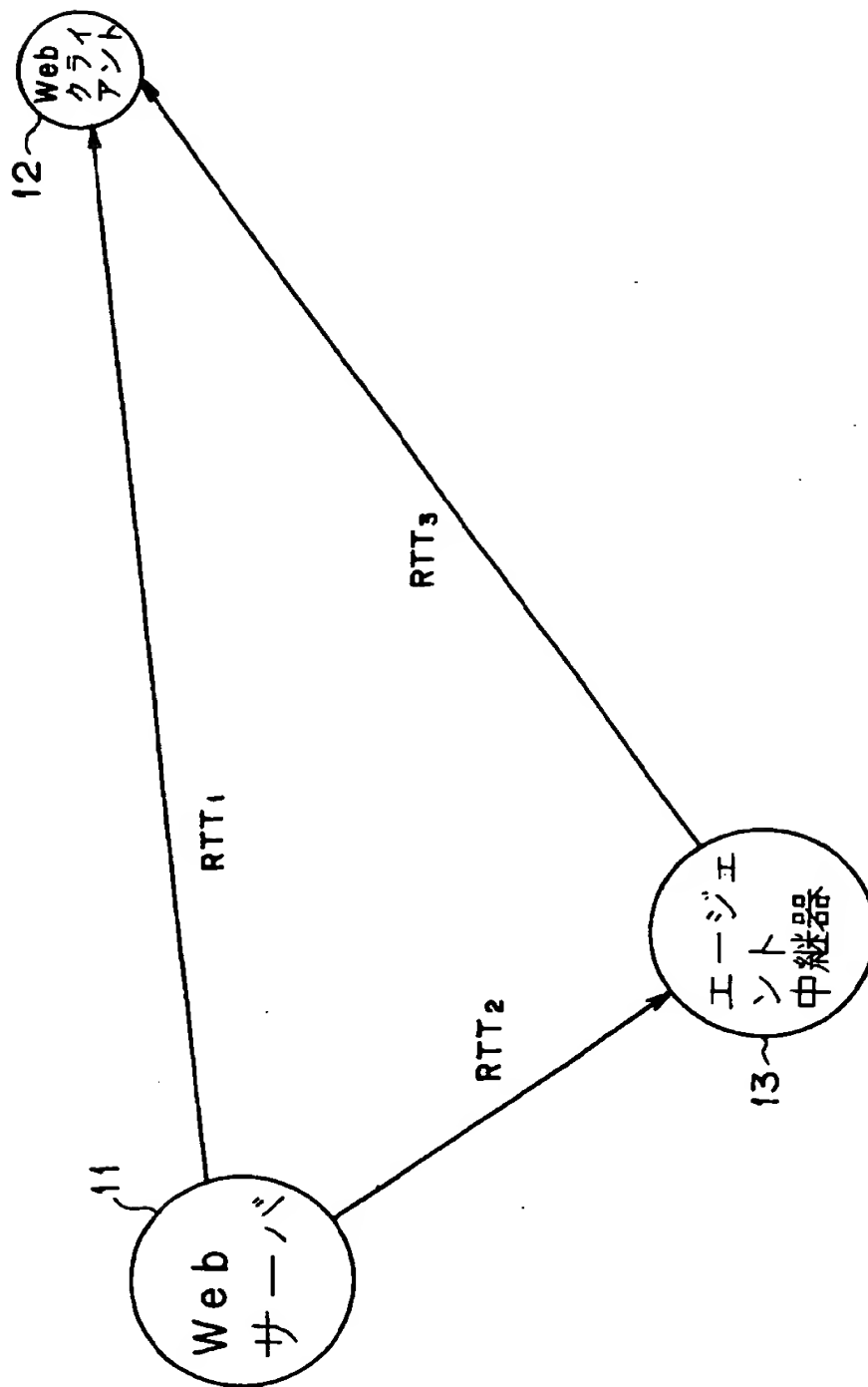
【図 1】

本 発 明 の 原 理 図



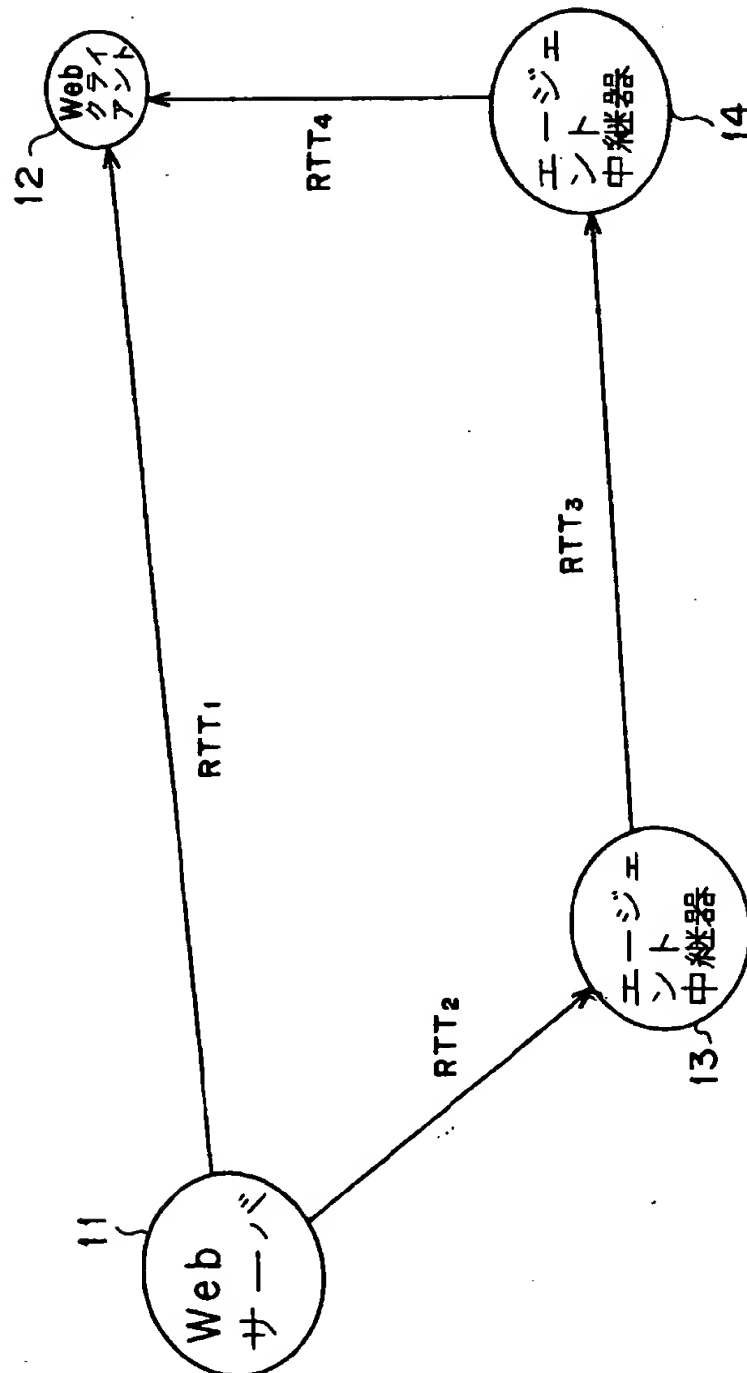
【図 2】

第1の通信システムの構成図



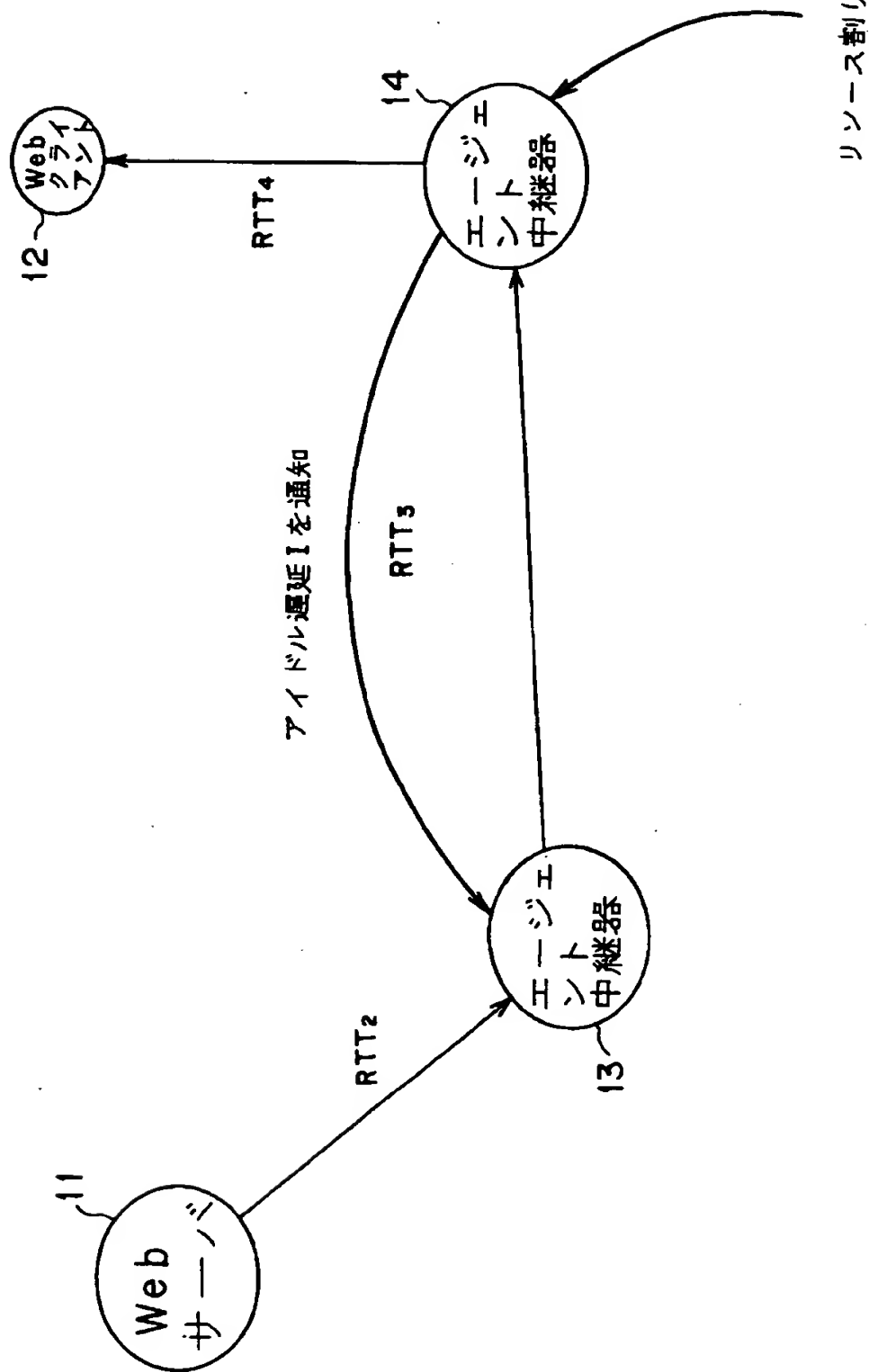
【図 3】

第2の通信システムの構成図



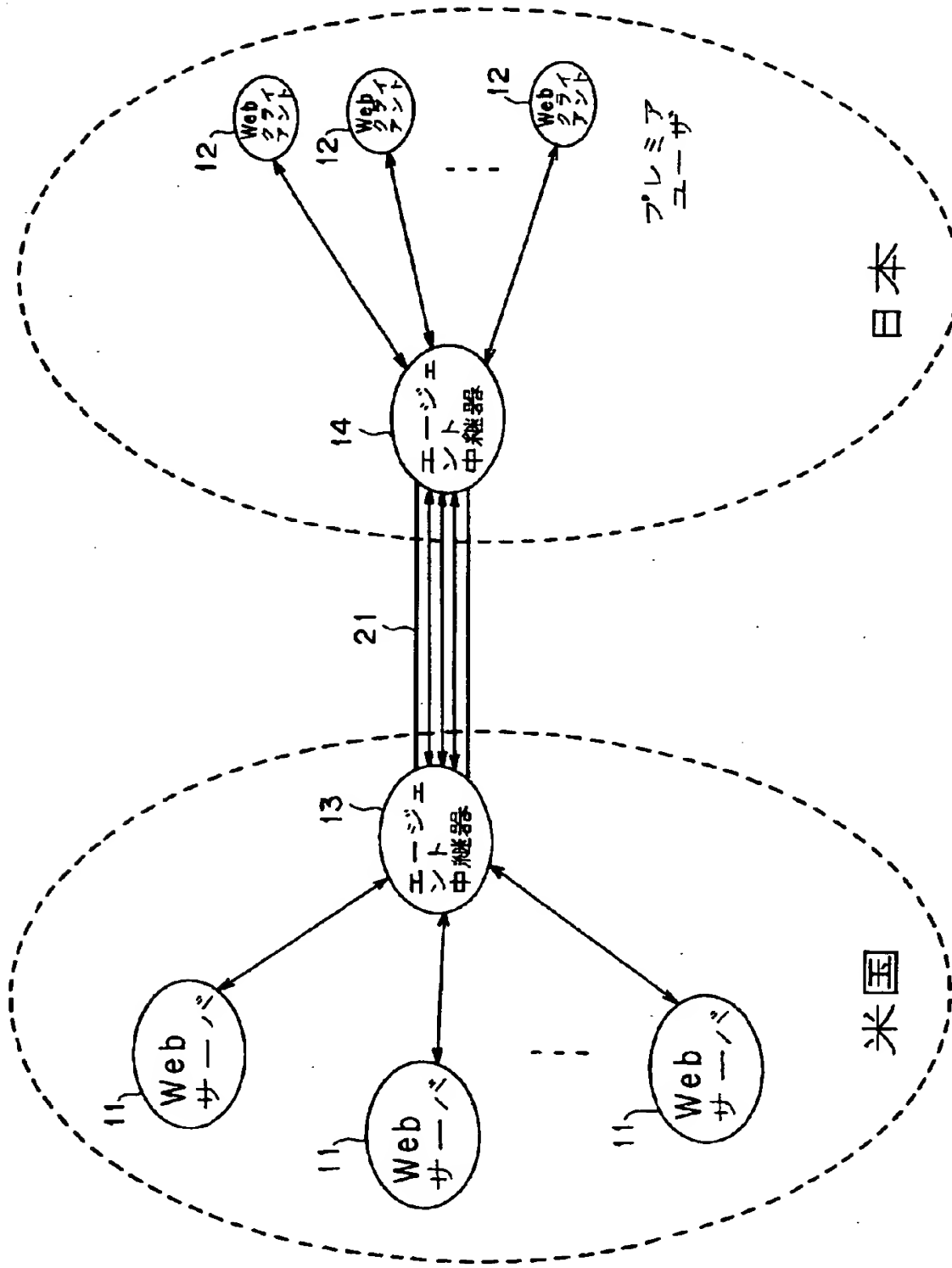
【図 4】

スロットリング制御を示す図



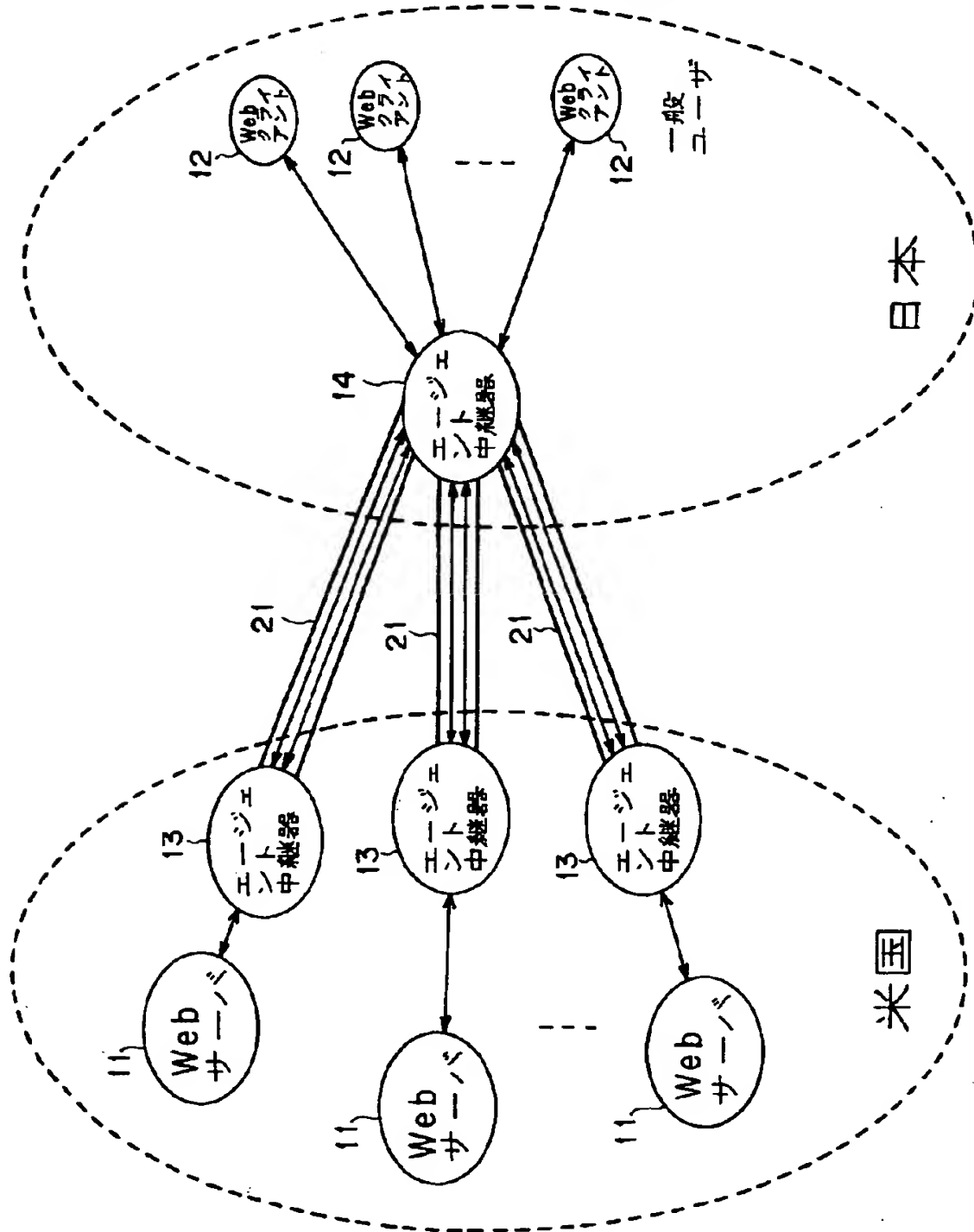
【図5】

プレミアムユーザーサービスを示す図



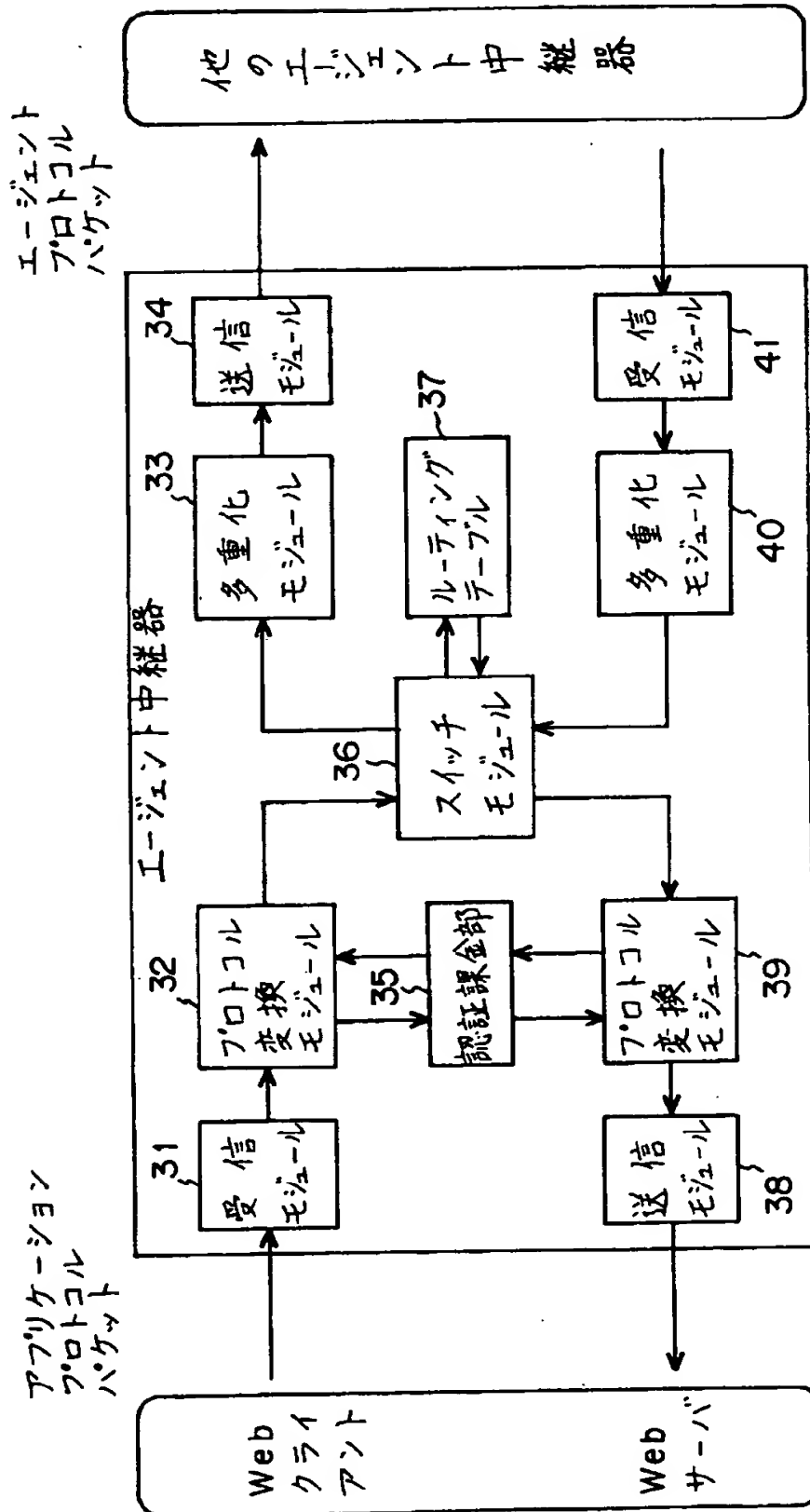
【図6】

クライアントミラーリングサービスを示す図



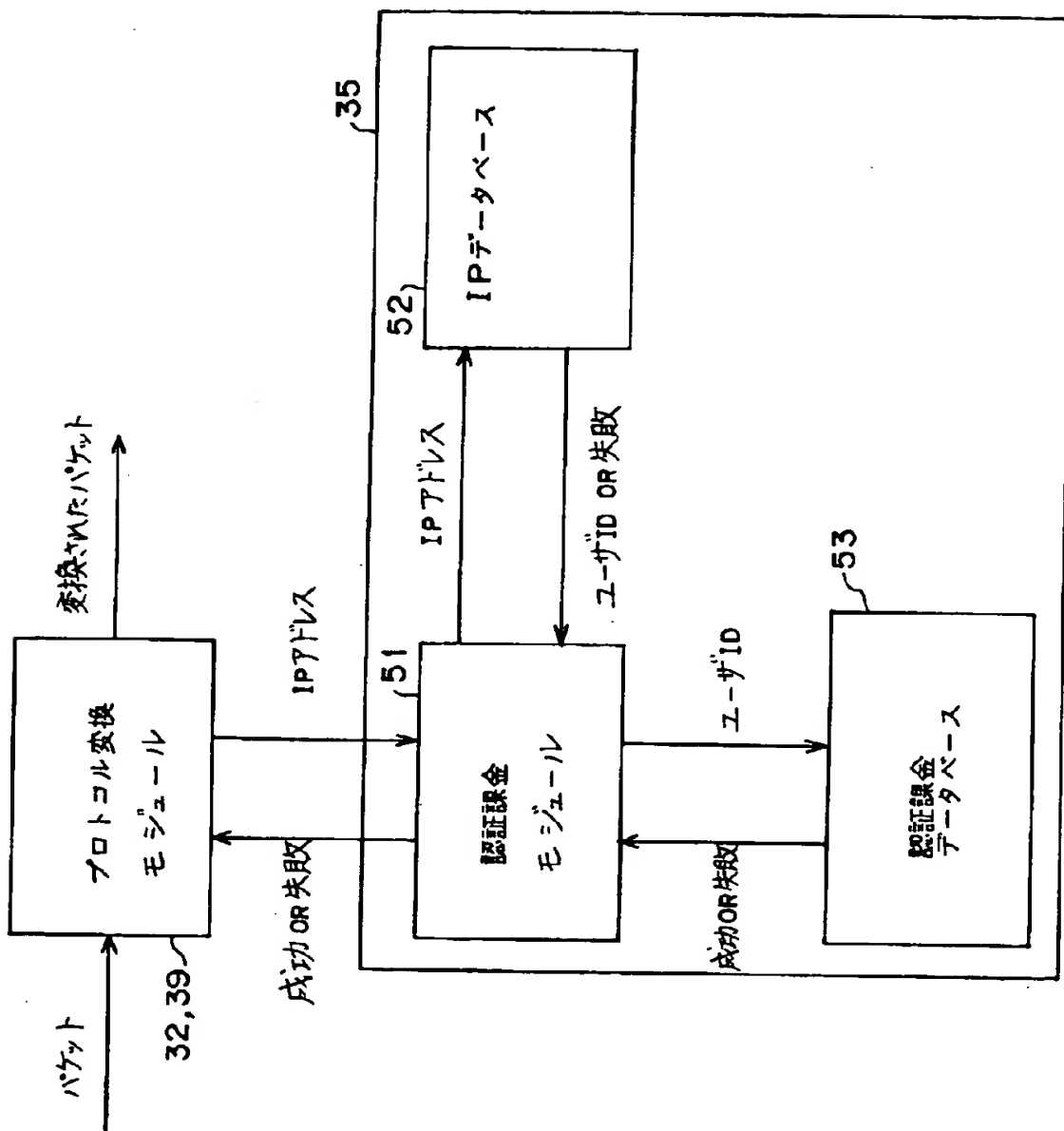
【図7】

第1のエージェント中継器の構成図



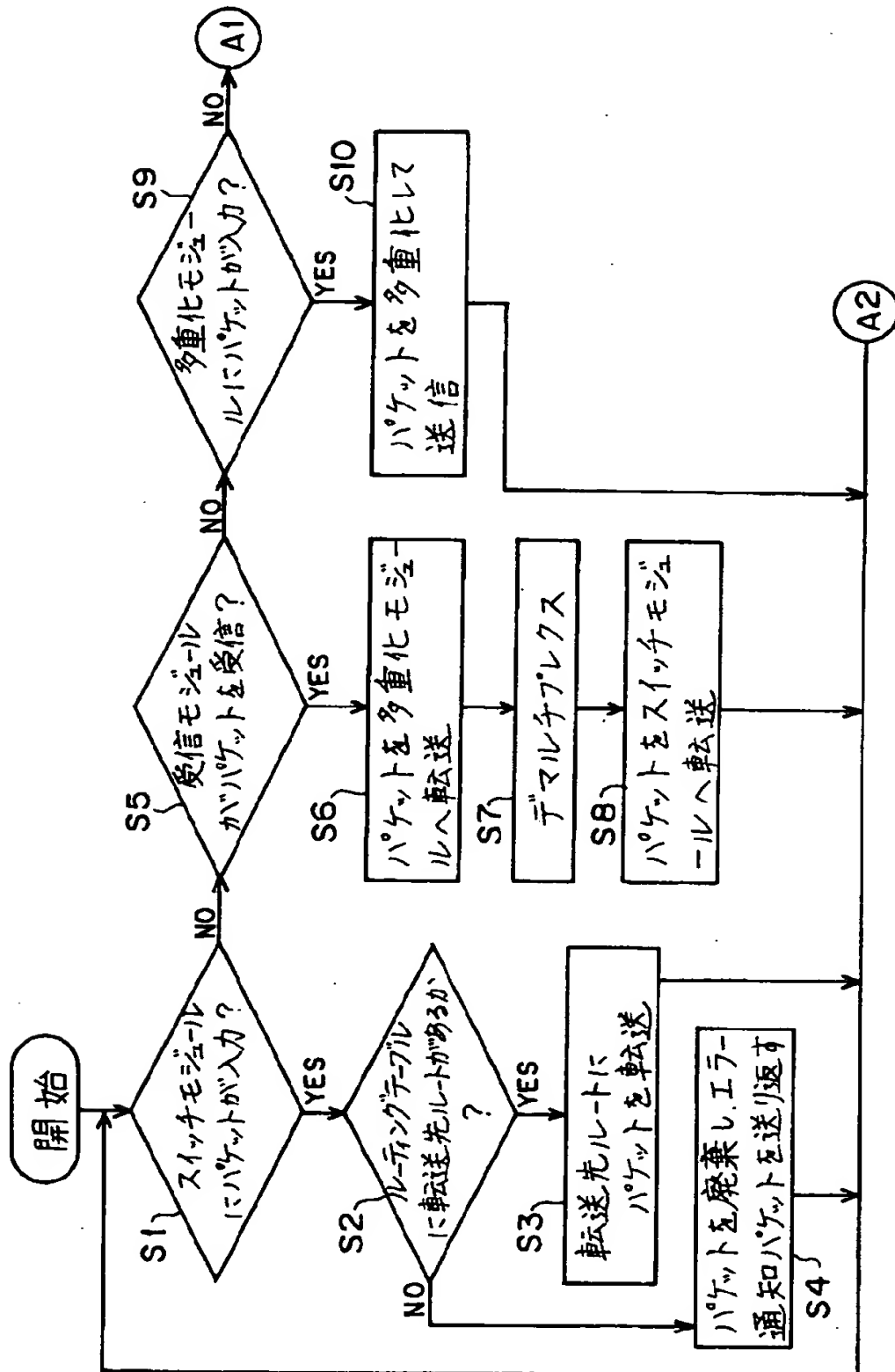
【図8】

認証課金部の構成図



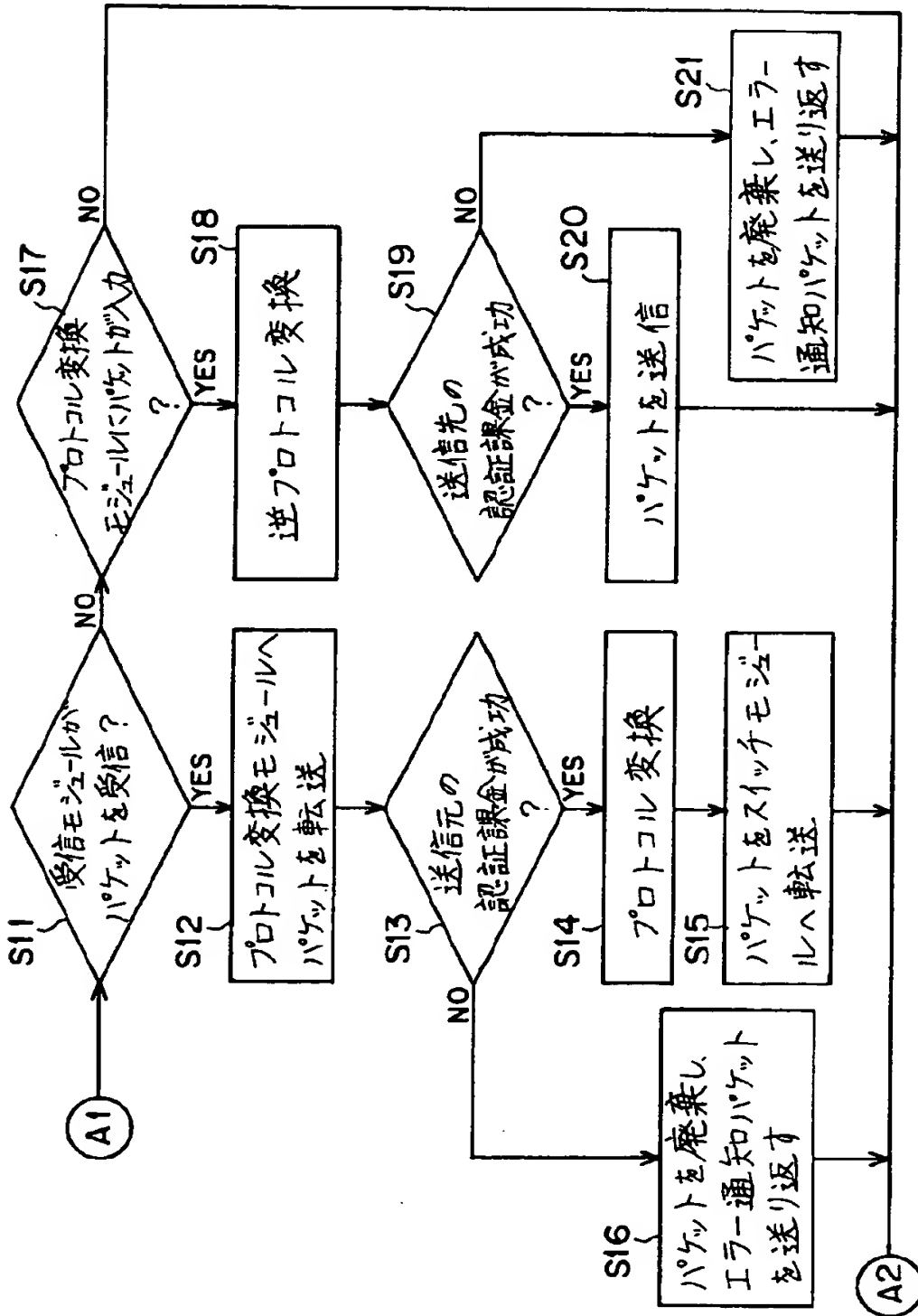
【図9】

エージェント中継器の処理のフローチャート(その1)



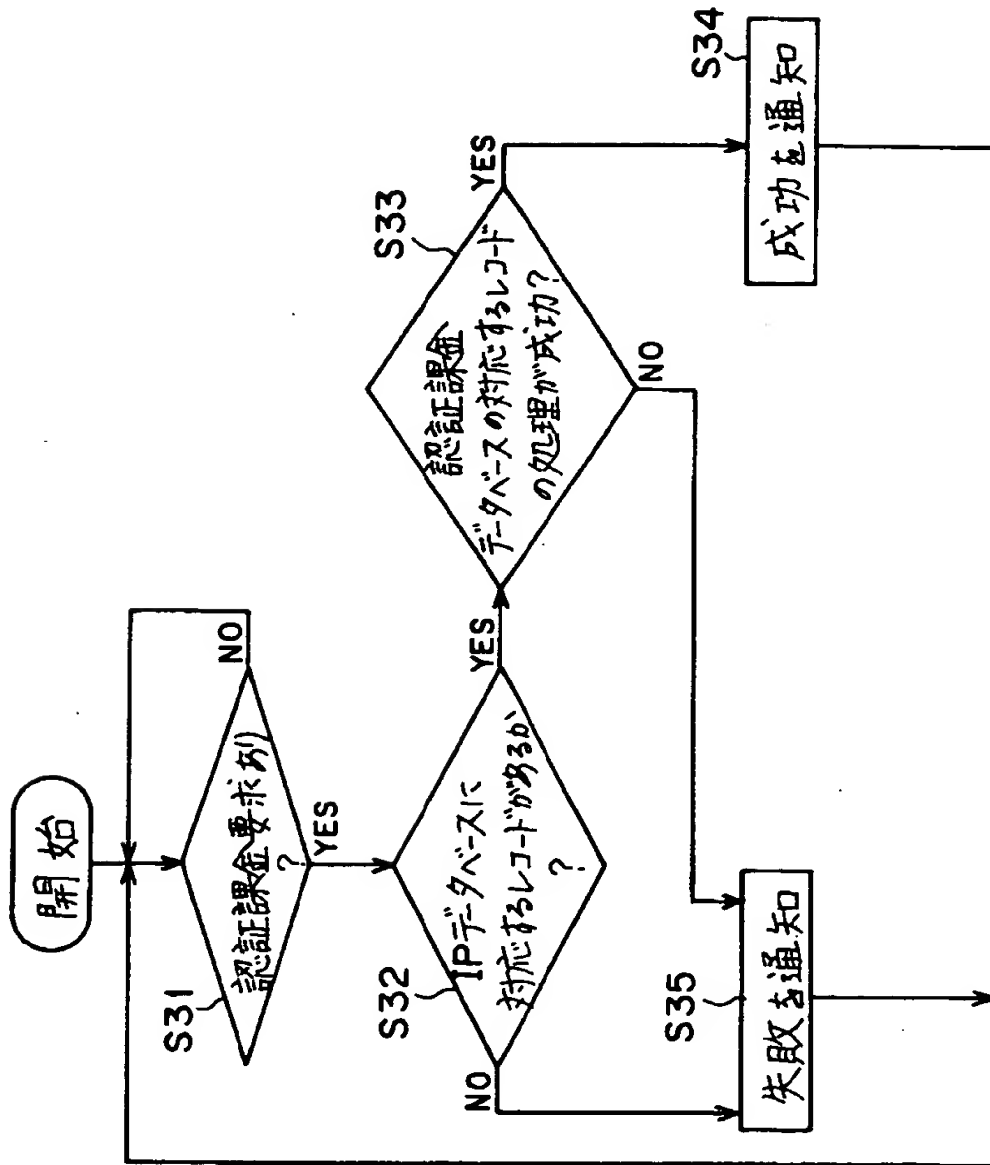
【図10】

エージェント中継器の処理のフローチャート(その2)



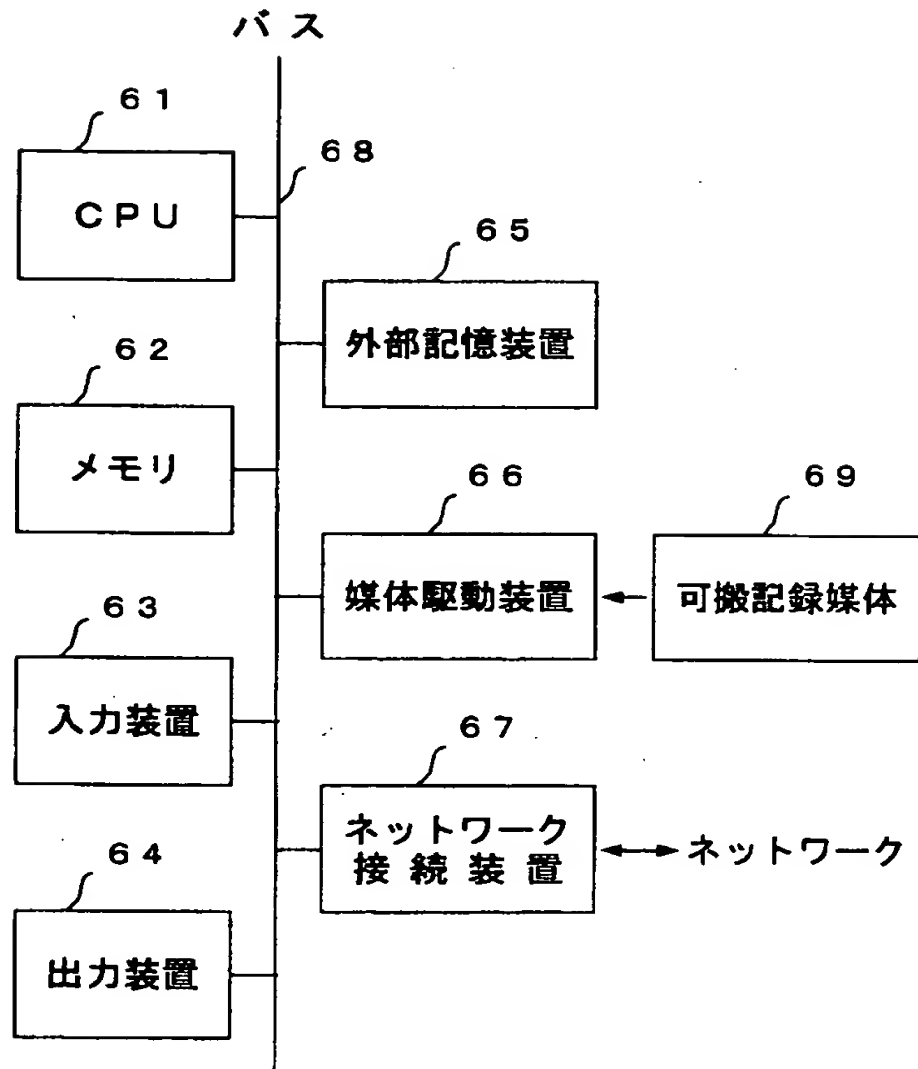
【図 1 1】

認証課金処理のフローチャート



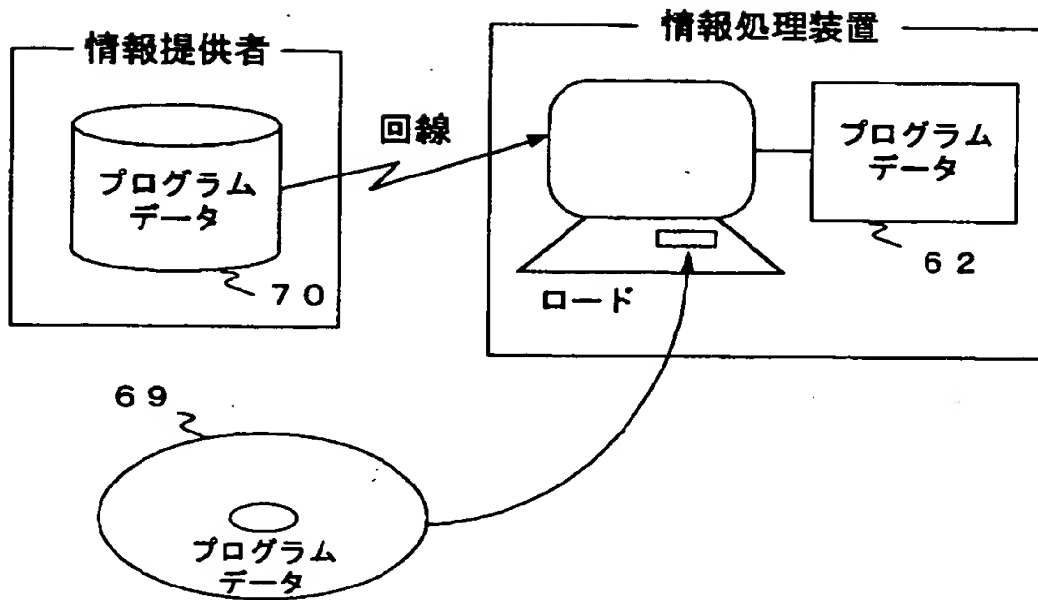
【図 1 2】

情 報 処 理 装 置 の 構 成 図



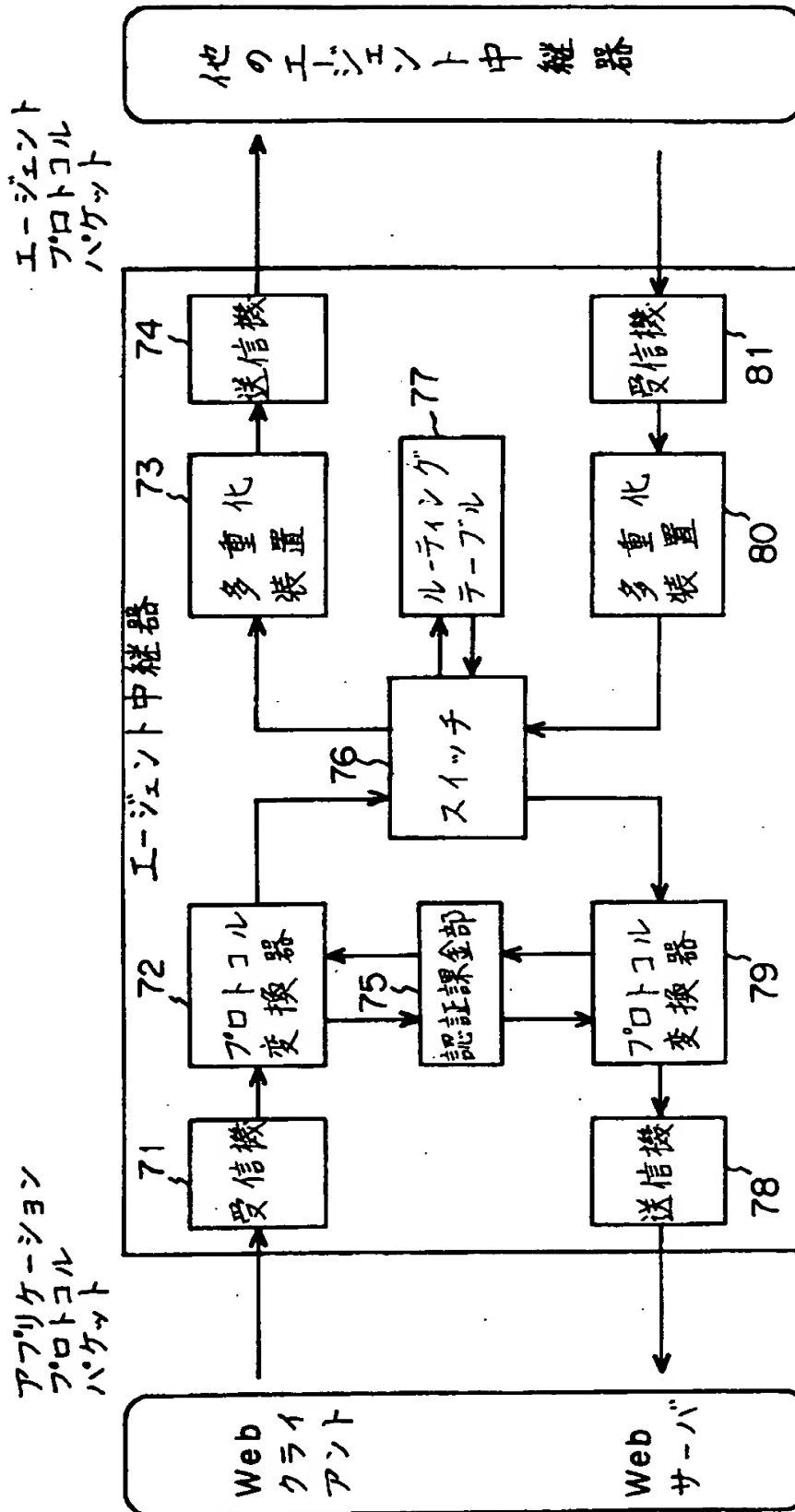
【図 1 3】

記 録 媒 体 を 示 す 図



【図 1 4】

第2のエージェント中継器の構成図



【書類名】 要約書

【要約】

【課題】 遅延の大きなネットワークにおいて、低コストでクライアント－サーバ間のスループットを向上させることが課題である。

【解決手段】 エージェント中継器 1 3 とエージェント中継器 1 4 の間で、サーバ 1 1 とクライアント 1 2 の間のアプリケーションプロトコルの接続を、巨大な通信ウィンドウを持つ多重化プロトコルに変換して、通信を中継する。また、エージェント中継器 1 3 が大きなバッファでサーバ 1 1 からのデータを受信することで、サーバ 1 1 がこの接続に割り当てるスループットを増大させる。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社